



HAL
open science

A batch sizing and scheduling problem on parallel machines with different speeds, maintenance operations, setup times and energy costs

Margaux Nattaf, Christian Artigues, Pierre Lopez, Rosa Medina, Lorena Pradenas, Victor Parada

► To cite this version:

Margaux Nattaf, Christian Artigues, Pierre Lopez, Rosa Medina, Lorena Pradenas, et al.. A batch sizing and scheduling problem on parallel machines with different speeds, maintenance operations, setup times and energy costs. International Conference on Industrial Engineering and Systems Management (IESM 2015), Oct 2015, Seville, Spain. 17p. hal-01166498

HAL Id: hal-01166498

<https://hal.science/hal-01166498>

Submitted on 22 Jun 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A batch sizing and scheduling problem on parallel machines with different speeds, maintenance operations, setup times and energy costs

Margaux Nattaf^{1,3}, Christian Artigues^{2,3}, Pierre Lopez^{2,3}, Rosa Medina⁴, Lorena Pradenas⁴, and Victor Parada⁵

¹Univ. de Toulouse, UPS, F-31400 Toulouse France, *Email: mnattaf@laas.fr*

²Univ. de Toulouse, LAAS, F-31400 Toulouse France, *Email: {artigues,lopez}@laas.fr*

³CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France

⁴Departamento de Ingeniería Industrial, Universidad de Concepción, Concepción, Chile, *Email: {lpradena,rosmedina}@udec.cl*

⁵Departamento de Ingeniería Informática, Universidad de Santiago de Chile, Santiago, Chile, *Email: victor.parada@usach.cl*

Abstract

This paper considers a production scheduling problem in a Chilean company from the metalworking industry. This company produces steel balls of different diameters on parallel production lines. There are two types of production lines and each production line may have a different speed for producing each diameter. Furthermore a setup time occurs when changing the diameter produced on each machine. Besides these production and setup operations, maintenance operations have to be scheduled. These machines are electrical and yield high energy demands. It is therefore crucial to minimize total energy consumption, which depends on batch/machine assignment and maximum demand on peak hours. We consider the batch sizing and scheduling problem involving electricity costs in a non-uniform parallel machine context. Given a demand for each family of steel balls, the problem consists in splitting the demand in sublots (batches) that have to be assigned and scheduled on the parallel machines together with the required maintenance operations. The goal is to complete the schedule before a common deadline while minimizing electricity costs. We propose to tackle this problem through mixed integer linear programming. We proposed a global formulation and a two-phase matheuristic. Computational results on realistic instances are provided.

1 Introduction

This paper considers a production scheduling problem in a Chilean company from the metalworking industry. The problem and the industrial context was described in [1]. This company produces steel balls on parallel production lines. The steel balls are obtained by roll forming or forging from a raw material consisting of metal bars. The steel balls are mainly used in the copper and gold

mining industry for mineral grinding, i.e. for reducing the size of the mineral particles to a maximal granularity that permits to remove the major part of impurities from the mineral. There are several type of steel balls to produce, each corresponding to a different ball diameter. There are two types of production lines : roll formers for small diameters and forges for larger diameters (although medium diameter balls can be produced by both production line types). Each production line may have a different speed for producing each diameter. Furthermore a setup time occurs when changing the diameter produced on each machine. Besides these production and setup operations, maintenance operations have to be scheduled. These machines are electrical and the production process results in high energy demands. As mentioned in [1], up to 50% of the production cost in such a manufacturing process can be due to electricity consumption. In Chile, as in many other places, different electricity rates are applied for peak hours and off-peak hours. So it can be crucial for metalworking companies to control the electricity demand during peak hours. Planning the maintenance and setup operations during the peak hours can be intuitively a policy that favors electricity cost decrease.

In this paper we consider a batch sizing and scheduling problem involving electricity costs in a non-uniform parallel machine context. Given a demand for each family of steel balls, the problem consists in splitting the demand in sublots (batches) that have to be assigned and scheduled on the parallel machines together with the required maintenance operations. The goal is to complete the schedule before a common deadline while minimizing electricity costs. We propose to tackle this problem through a mixed integer linear programming approach.

Section 2 presents the batch sizing and scheduling problem. Section 3 is devoted to a brief presentation of the related work. Section 4 gives the proposed mixed-integer linear programming formulation. Section 5 gives a simplified MILP formulation that ignores the peak costs and that serves as a basis for a matheuristic. Section 6 presents the considered realistic problem instances and the results obtained by our approach. Section 7 draws concluding remarks and directions for future work.

2 The industrial scheduling problem

The problem involves a set \mathcal{J} of n lots (jobs) to be scheduled on a set \mathcal{M} of m machines. There is total a production demand D_j for each job $j \in \mathcal{J}$ to be fulfilled during the scheduling horizon given by a time interval $[0, T]$. Each job $j \in \mathcal{J}$ can be split into a maximum number of $\lfloor D_j/\epsilon \rfloor$ sublots (batches), where ϵ is the minimum batch size. The machines able to produce a job j are gathered in set \mathcal{M}_j . For a given job $j \in \mathcal{J}$, a machine $k \in \mathcal{M}_j$ has a production speed v_j^k . Whenever a batch of a job $j \in \mathcal{J}$ is scheduled immediately after a batch of job $i \in \mathcal{J}$ on a machine $k \in \mathcal{M}$, a setup time $s_{ij}^k \geq 0$ is necessary on the machine. Each machine is assumed to have a constant electrical power demand w_k . There are H peak periods within the horizon, each being defined by an interval $[a_h, f_h)$ with $a_h < f_h$ for $h = 1, \dots, H$ and $f_h < a_{h+1}$ for $h = 1, \dots, H - 1$. Concerning cost minimization, we considered a weighted sum problem where α denotes the weight of the total energy consumption while β denotes the weight of the maximum consumption during peak hours.

A solution consists in determining:

- A set of batches \mathcal{B}_j with $|\mathcal{B}_j| = B_j \geq 1$, for each job $j \in \mathcal{J}$,
- A production amount $q_{j,b} \geq \epsilon$, for each batch $b \in \mathcal{B}_j$.
- A number A_k of batches assigned to machine $k \in \mathcal{M}$, defining a set of positions $\mathcal{L}_k = \{1, \dots, A_k\}$,
- A batch $(j_{k,l}, b_{k,l})$ assigned to each pair (machine, position), for each machine $k \in \mathcal{M}$ and each position $l \in \mathcal{L}_k$.
- A start time $S_{j,b}$, for each job $j \in \mathcal{J}$ and for each batch $b \in \mathcal{B}_j$.

Let $\mathcal{B} = \cup_{j \in \mathcal{J}} \{(j, b) | b \in \mathcal{B}_j\}$ denote the set of pairs (job, batch index). Let $\mathcal{A}_k = \{(j_{k,l}, b_{k,l}) | l \in \mathcal{L}_k\}$ the set of batches assigned to machine $k \in \mathcal{M}$.

The problem constraints can be stated as follows:

One and only one machine has to be assigned to each batch.

$$C \left(\sum_{k \in \mathcal{M}} A_k = |\mathcal{B}| \right) \wedge \left(\bigcup_{k \in \mathcal{M}} \mathcal{A}_k = \mathcal{B} \right) \quad (1)$$

A batch can only be assigned on an authorized machine.

$$Ck \in \mathcal{M}_j \quad \forall k \in \mathcal{M}, \forall j \in \{i \in \mathcal{J} | (i, b) \in \mathcal{A}_k\} \quad (2)$$

The demand must be satisfied for each job.

$$C \sum_{b \in \mathcal{B}_j} q_{j,b} = D_j \quad \forall j \in \mathcal{J} \quad (3)$$

The duration of a batch is equal to its production divided by the speed of its assigned machine.

$$Cp_{j_{k,l}, b_{k,l}} = q_{j_{k,l}, b_{k,l}} / v_{j_{k,l}}^k \quad \forall k \in \mathcal{M}, \forall l \in \mathcal{L}_k \quad (4)$$

No batches assigned to the same machine may overlap and the setup time between two consecutive batches on the same machine must be respected.

$$CrS_{j_{k,l+1}, b_{k,l+1}} \geq S_{j_{k,l}, b_{k,l}} + p_{j_{k,l}, b_{k,l}} + s_{j_{k,l}, j_{k,l+1}}^k \quad \forall k \in \mathcal{M}, \forall l \in \mathcal{L}_k \setminus \{L_k\} \quad (5)$$

All production must fit in the time horizon.

$$CS_{j,b} + p_{j,b} \leq T \quad \forall (j, b) \in \mathcal{B} \quad (6)$$

We need not introduce maintenance operations in the model. We consider that there is a subset \mathcal{J}^M of the jobs \mathcal{J} that correspond to maintenance operations. They concern a single machine ($|\mathcal{M}_j| = 1$). The maintenance duration is given by D_j and we have $v_j^k = 1$ for $k \in \mathcal{M}_j$. We denote by \mathcal{B}^M the set of batches corresponding to maintenance operations. We introduce last the constraints that each maintenance job cannot be splitted:

$$C|\mathcal{B}_j| = 1 \quad \forall j \in \mathcal{J}^M \quad (7)$$

We assume that there is no required setup time between a maintenance job and a production job. However we make the realistic assumption that the maintenance duration is such that $D_j \geq s_{ii'}$, $\forall j \in \mathcal{J}^M$, $\forall i, i' \in \mathcal{J} \setminus \mathcal{J}^M$, $i \neq i'$, i.e. is larger than any other setup time. Furthermore we assume that all other setup times satisfy the triangle inequality

$$Cs_{ij}^k \geq s_{il}^k + s_{lj}^k \quad \forall k \in \mathcal{M}, \forall i, j, l \in \mathcal{J} \setminus \mathcal{J}^M, i \neq j \neq l$$

It follows that the triangle inequality is respected.

A solution is feasible if and only if it satisfies constraints 1–7.

There are two components linked to energy consumption. As the machines have different speeds and different power consumption, the first part of the objective is to minimize the total electricity consumption C .

$$CC = \sum_{k \in \mathcal{M}} w_k \sum_{(j,b) \in \mathcal{A}_k \setminus \mathcal{B}^M} p_{j,b} \quad (8)$$

We are also interested in the maximum demand over the peak periods. Let us define for convenience the set of batches in process at a given time point t

$$CB(t) = \{(j, b) \in \mathcal{B} | S_j, b \leq t < S_{j,b} + p_{j,b}\} \quad (9)$$

Then the maximum demand over peak hours is equal to

$$CP_{\max} = \max_{h \in \mathcal{H}} \max_{t \in [a_h, b_h)} \sum_{k \in \mathcal{M}, \exists (j,b) \in CB(t) \cap \mathcal{A}_k \setminus \mathcal{B}^M} w_k \quad (10)$$

Remark that maintenance operations are excluded in the computation of these costs. Then the problem objective can be written

$$C \min \alpha C + \beta P_{\max} \quad (11)$$

Note that the problem is NP-hard as the problem of finding a solution has the decision variant of the parallel machine problem as special case. The above-defined constraints can be used to check the feasibility and cost of a solution in polynomial time but they cannot be implemented as is in a mathematical programming solver.

We present a small example and its optimal solution. Consider $n = 3$ jobs on $m = 2$ lines with demands $D_1 = 10$, $D_2 = 12$, $D_3 = 6$. There is one peak period $h = 1$ with $a_h = 2$ and $b_h = 4$ and a common deadline of $T = 6$ for all jobs and maintenance operations. There are two maintenance operations $\mathcal{J}^M = \{4, 5\}$ of duration $D_4 = 2$ on machine 1 and $D_5 = 1$ on machine 2 to be performed also before the deadline. Machine power demands are $w_1 = w_2 = 10$. For each job, authorized machines are $\mathcal{M}_1 = \{1, 2\}$, $\mathcal{M}_2 = \{1\}$, $\mathcal{M}_3 = \{2\}$, $\mathcal{M}_3 = \{2\}$, $\mathcal{M}_4 = \{1\}$, $\mathcal{M}_5 = \{2\}$. Setup and speed matrices are respectively equal to

$$(s_{i,j}^1)_{i,j \in \mathcal{J}} = (s_{i,j}^2)_{i,j \in \mathcal{J}} = \begin{bmatrix} 0 & 2 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

$$(v_j^k)_{j \in \mathcal{J}, k \in \mathcal{M}} = \begin{bmatrix} 2 & 3 \\ 6 & 0 \\ 0 & 2 \end{bmatrix}$$

Figure 1 presents an optimal solution for this example where $\mathcal{B}_1 = \{1, 2\}$ (job 1 is split into two batches), $\mathcal{B}_2 = \mathcal{B}_3 = 1$ (jobs 2 and 3 are scheduled in one batch). Batch machine assignment and positions are given by $\mathcal{A}_1 = \{(j_{1,1}, b_{1,1}) = (1, 1), (j_{1,2}, b_{1,2}) = (2, 1)\}$ and $\mathcal{A}_2 = \{(j_{2,1}, b_{2,1}) = (1, 2), (j_{2,2}, b_{2,2}) = (3, 1)\}$. Productions inside each batch are given by $q_{1,1} = 4, q_{1,2} = 6, q_{2,1} = 12, q_{3,1} = 6$. According to the assigned machine speeds, the batch duration are given by $p_{1,1} = q_{1,1}/v_1^1 = 2, p_{1,2} = q_{1,2}/v_1^2 = 2, p_{2,1} = q_{2,1}/v_2^1 = 2, p_{3,1} = q_{3,1}/v_3^2 = 3$. The start times of the different batches and of the maintenance operations (in gray) are displayed in the Gantt chart. According to the machine power demand, total consumption of machine 1 is $4 \times w_1 = 40$ and total consumption of machine 2 is $5 \times 10 = 50$. Hence we obtain a total energy consumption of 90 and a maximal demand on peak hours of 10. We observe that batch splitting has been necessary to meet the deadline and minimize energy consumption, while maintenance has been used to avoid large demands during peak hours.

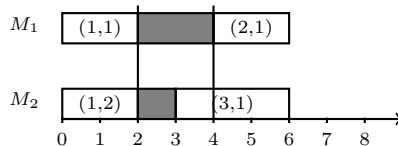


Figure 1: Optimal solution of a simple instance

Hence, after a brief presentation of the related work, we will introduce a mixed-linear integer programming formulation.

3 Related work and modeling issues

A vast literature considers upper level lot-sizing and scheduling problems in parallel machine environment with setup times (such as in [2, 3]) involving, among others, inventory costs and also using discrete time mixed-integer linear programming formulations. First, the discrete time models do not fit our problem as the duration of the subplot is a function of the subplot size (a continuous variable) and the machine speed for the family of balls corresponding to the subplot. Furthermore we do not have holding costs in our model. More generally, our problem is in fact related to the (operational) production scheduling level where it is assumed that lots of products aiming at satisfying the demand in a given period have already been constituted at the (tactical) planning level.

Lot streaming (also called lot splitting) is a known technique to improve the performance of production scheduling environment at the operational level [4, 5] by allowing to split the jobs across the different production stages or lines. In our case, the lot streaming process is simplified by the fact that we have a single stage. This is also the reason why we rather use the “batch sizing” term rather than lot streaming or splitting (we borrowed this term from [6]).

The possibility of splitting the jobs (lots/batches) adds another dimension to the decisions besides assignment of the jobs to the machine and sequencing/scheduling of the job operations. Many lot streaming/batch sizing and scheduling problems have been considered in the literature for different scheduling context, including parallel machine scheduling for which efficient algorithms are available [7, 8]. However, these first studies did not consider the presence

of setup times that can be needed on a machine between two sublots. This feature, that occurs in our case when the ball diameter is changed, puts obviously a limit to the interest of systematic job splitting. Several approaches can be found for parallel machine lot streaming/batch sizing problems with setup times [9, 10, 11]. Compared to these previous works, our study is concerned with electricity cost minimization objectives. In manufacturing, energy constraints and costs are becoming crucial. Consequently, energy-related objectives have been recently considered both in lot-sizing problems [12] and in production scheduling problems [13, 14, 15]. However to our knowledge, reducing the energy consumption cost by an integrated management of lot streaming/batch sizing, setup and maintenance scheduling in a parallel machine scheduling environment is a novel approach.

In this paper, one of the objective is to come up with a mixed-integer linear programming formulation (MILP) of the problem. In scheduling problems, there are standardly three categories of MILP formulations [16]: the continuous time formulations with sequencing variables, the continuous time formulations with positional (or event-based) variables and the discrete time formulations. In the considered problem, the continuous lot splitting possibility and the variable machine speed combined with large time horizons would render a discrete time formulation (such as the ones generally used in the lot-sizing literature) impractical. Therefore we need a continuous time formulation. Now looking at the conceptual continuous time formulation (1–11), most non-linear constraints could be linearized in a standard way yielding a continuous time (also called disjunctive) formulation with sequencing variables $x_{j,b,i,b'} \in \{0, 1\}$ where $x_{j,b,i,b'} = 1$ if and only if batch (i, b') is scheduled after batch (j, b) on the same machine. In [17], the authors consider such a formulation for a simple identical parallel machine scheduling problem with job-splitting and sequence-dependent setup times with the makespan criterion. In this simple case splitting a job is only useful if all sublots are scheduled on different machines, hence the authors consider binary sequencing variables of the type $x_{i,j,k}$ where $x_{i,j,k} = 1$ is the subplot of job i in machine k is sequenced before the subplot of job j in machine k . However in our case, it can be necessary to have two batches of the same job on the same machine, especially due to the interest of avoiding production during the peak hours. Among many other configurations Fig. 2 shows an optimal solution for a 1 machine and 3 jobs example ($m = 1$ and $n = 3$) with setup times $s_{12} = s_{21} = s_{13} = s_{31} = 1$, $s_{23} = s_{32} = 2$, unit machine speed, unit machine power, demands $D_1 = 2$, $D_2 = D_3 = 1$, deadline $T = 8$ and weights $\alpha = 0$, $\beta = 1$. Setup times are shown in gray. The optimal solution (of cost 0) needs to split job 1 in two batches to have no production within the peak hour (interval marked with a P).

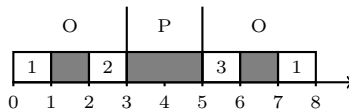


Figure 2: Two batches of a job on the same machine

Another issue for modeling the problem in continuous time is the linearization of the expressions (9, 10) defining the maximum demand over the contin-

uous horizon. An important remark is that the electricity consumption only changes at a batch start or end time. This clearly brings us to consider an event-based model. These models consider a finite number of events and use an event-indexed continuous variable to associate a start time to each event and a binary variable to assign a batch start or end time. Consequently, we can also associate a continuous variable representing the total electricity consumption at a given event, that shall not change until the next event. Note that event-based models were previously used for lot streaming and scheduling in hybrid flow-shops [18] and are also widely used in batch scheduling in the process industry [19]. They were also more recently introduced in resource-constrained project scheduling problems [20] and in scheduling problems under energy constraints [21].

4 A continuous time event-based mixed-integer linear programming formulation

The event-based model we propose for the problem, is based on the fact that energy consumption can only change at the beginning or at the end of each batch. Hence we fix for each job its maximum number of batches B_j and we consider a set \mathcal{E} of $2 \sum_{j=1}^n B_j$ events. Now the start-end event-based formulation of the problem involves the following decision variables (variables with domain $[0, 1]$ are continuous variables that will be set binary by the constraints involving other binary variables).

- $t_e \geq 0$ is the time of event $e \in \mathcal{E}$.
- $x_{j,b}^e \in \{0, 1\}$ is equal to 1 iff batch $(j, b) \in \mathcal{B}$ starts at event $e \in \mathcal{E}$.
- $y_{j,b}^e \in \{0, 1\}$ is equal to 1 iff batch $(j, b) \in \mathcal{B}$ ends at event $e \in \mathcal{E}$.
- $a_{j,b}^k \in \{0, 1\}$ is equal to 1 iff batch $(j, b) \in \mathcal{B}$ is assigned to machine $k \in \mathcal{M}$.
- $x_{j,b}^{e,k} \in [0, 1]$ auxiliary variable equal to product $x_{j,b}^e a_{j,b}^k$.
- $y_{j,b}^{e,k} \in [0, 1]$ auxiliary variable equal to product $y_{j,b}^e a_{j,b}^k$.
- $o_{e,k} \in [0, 1]$ is equal to 1 iff machine $k \in \mathcal{M}$ is on at event $e \in \mathcal{E}$.
- $p_{j,b} \geq 0$ is the production of batch $(j, b) \in \mathcal{B}$.
- $P_{e,h} \in \{0, 1\}$ that models the relative positioning of intervals $[t_e, t_{e+1})$ and peak hour interval $[a_h, f_h)$ in the sense that $P_{e,h} = 0 \Rightarrow t_e \geq f_h$.
- $P_{h,e} \in \{0, 1\}$ that models the relative positioning of intervals $[t_e, t_{e+1})$ and peak hour interval $[a_h, f_h)$ in the sense that $P_{h,e} = 0 \Rightarrow t_e + 1 \leq a_h$.
- $P_e^k \in [0, 1]$ event e is active on a peak hour on machine k (i.e. $[t_e, t_{e+1})$ has a non-empty intersection with a peak hour interval and machine k is on at event e).
- $W_{j,b} \geq 0$ total electrical consumption of batch (j, b) .

The constraints can be expressed as follows. M denotes a sufficiently large integer.

$$Ct_e \leq t_{e+1} \quad \forall e \in \mathcal{E} \setminus \{|\mathcal{E}|\} \quad (12)$$

Each batch (j, b) has to be scheduled at most once.

$$C \sum_{e \in \mathcal{E}} x_{j,b}^e \leq 1 \quad \forall (j, b) \in \mathcal{B} \quad (13)$$

$$C \sum_{e \in \mathcal{E}} y_{j,b}^e \leq 1 \quad \forall (j, b) \in \mathcal{B} \quad (14)$$

A started batch must be ended, and reciprocally.

$$C \sum_{e \in \mathcal{E}} y_{j,b}^e = \sum_{e \in \mathcal{E}} x_{j,b}^e \quad \forall (j, b) \in \mathcal{B} \quad (15)$$

The end event of a batch cannot be lower than its start event.

$$C \sum_{f \in \mathcal{E}, f \geq e} x_{j,b}^f \geq \sum_{f \in \mathcal{E}, f \geq e} y_{j,b}^f \quad \forall (j, b) \in \mathcal{B}, \forall e \in \mathcal{E} \quad (16)$$

A batch has to be assigned to a machine to have a non-zero production (as B_j is the maximum number of batch for job j , there can be some unassigned batches).

$$Cp_{j,b} \leq \sum_{k \in \mathcal{M}_j} a_{j,b}^k D_j \quad \forall (j, b) \in \mathcal{B} \quad (17)$$

A batch assigned to a machine must have a minimum size.

$$Cp_{j,b} \geq \sum_{k \in \mathcal{M}_j} a_{j,b}^k \epsilon \quad \forall (j, b) \in \mathcal{B} \quad (18)$$

A batch has to be assigned to at most one machine among the candidate machines.

$$C \sum_{k \in \mathcal{M}_j} a_{j,b}^k \leq 1 \quad \forall (j, b) \in \mathcal{B} \quad (19)$$

The following constraints are set on auxiliary variables $x_{j,b}^{e,k}$ and $y_{j,b}^{e,k}$ to linearize products $x_{i,b}^e a_{j,b}^k$ and $y_{i,b}^e a_{j,b}^k$.

$$Cx_{j,b}^{e,k} \geq x_{j,b}^e + a_{j,b}^k - 1 \quad \forall (j, b) \in \mathcal{B}, \forall e \in \mathcal{E}, \forall k \in \mathcal{M} \quad (20)$$

$$Cx_{j,b}^{e,k} \leq x_{j,b}^e \quad \forall (j, b) \in \mathcal{B}, \forall e \in \mathcal{E}, \forall k \in \mathcal{M} \quad (21)$$

$$Cx_{j,b}^{e,k} \leq a_{j,b}^k \quad \forall (j, b) \in \mathcal{B}, \forall e \in \mathcal{E}, \forall k \in \mathcal{M} \quad (22)$$

$$C y_{j,b}^{e,k} \geq y_{j,b}^e + a_{j,b}^k - 1 \quad \forall (j,b) \in \mathcal{B}, \forall e \in \mathcal{E}, \forall k \in \mathcal{M} \quad (23)$$

$$C y_{j,b}^{e,k} \leq x_{j,b}^e \quad \forall (j,b) \in \mathcal{B}, \forall e \in \mathcal{E}, \forall k \in \mathcal{M} \quad (24)$$

$$C y_{j,b}^{e,k} \leq a_{j,b}^k \quad \forall (j,b) \in \mathcal{B}, \forall e \in \mathcal{E}, \forall k \in \mathcal{M} \quad (25)$$

A machine is on at an event e depending on its status at the preceding event and/or start or completion of a task on this machine (we define also $o_{0,k} = \sum_{(j,b) \in \mathcal{B}} x_{j,b}^{0,k}$, $\forall k \in \mathcal{M}$).

$$\begin{aligned} Cro_{e,k} &= o_{e-1,k} - \sum_{(j,b) \in \mathcal{B}} y_{j,b}^{e,k} \\ &\quad + \sum_{(j,b) \in \mathcal{B}} x_{j,b}^{e,k} \quad \forall e \in \mathcal{E}, \forall k \in \mathcal{M} \end{aligned} \quad (26)$$

The common deadline must be satisfied.

$$Ct_{|\mathcal{E}|} \leq T \quad (27)$$

Setup times must be satisfied between two consecutive batches.

$$\begin{aligned} Ct_f &\geq t_e + s_{ij}^k (x_{j,b}^{f,k} + y_{i,b'}^{e,k} - 1) \quad \forall e, f \in \mathcal{E}, f \geq e, \\ &\quad \forall (j,b), (i,b') \in \mathcal{B}, (j,b) \neq (i,b'), \forall k \in \mathcal{M} \end{aligned} \quad (28)$$

Batches can be assigned only on authorized machines.

$$C a_{j,b}^k = 0 \quad \forall (j,b) \in \mathcal{B}, \forall k \in \mathcal{M} \setminus \mathcal{M}_j \quad (29)$$

End and start time events of a batch have to be spaced according to the batch duration on the assigned machine.

$$\begin{aligned} Ct_f &\geq t_e + p_{j,b}/v_{j,k} - M(2 - x_{j,b}^{e,k} - y_{j,b}^{f,k}) \\ &\quad \forall e, f \in \mathcal{E}, f \geq e, \forall (j,b) \in \mathcal{B}, \forall k \in \mathcal{M} \end{aligned} \quad (30)$$

$$\begin{aligned} Ct_f &\leq t_e + p_{j,b}/v_{j,k} + M(2 - x_{j,b}^{e,k} - y_{j,b}^{f,k}) \\ &\quad \forall e, f \in \mathcal{E}, f \geq e, \forall (j,b) \in \mathcal{B}, \forall k \in \mathcal{M} \end{aligned} \quad (31)$$

The demand of each job must be satisfied.

$$C \sum_{b \in \mathcal{B}_j} p_{j,b} = D_j \quad \forall j \in \mathcal{J} \quad (32)$$

The following constraints set the relative positioning of event e and peak hour period h , in the sense that if $P_{e,h} = 0$, intersection of $[t_e, t_e + 1)$ and peak hour interval $[a_h, f_h)$ is empty because $t_e \geq f_h$

$$Ct_e \geq f_h(1 - P_{e,h}) \quad \forall e \in \mathcal{E}, \forall h \in \mathcal{H} \quad (33)$$

The following constraints set the relative positioning of event e and peak hour period h , in the sense that if $P_{h,e} = 0$, intersection of $[t_e, t_e + 1)$ and peak hour interval $[a_h, f_h)$ is empty because $t_e + 1 \leq a_h$)

$$Ct_{e+1} - a_h \leq TP_{h,e} \quad \forall e \in \mathcal{E}, \forall h \in \mathcal{H} \quad (34)$$

Machine k is marked as active during peak hour at event e whenever machine k is on at event e and interval $[t_e, t_{e+1})$ intersects peak hour interval $[a_h, f_h)$.

$$\begin{aligned} CP_e^k &\geq P_{e,h} + P_{h,e} + o_{e,k} - 2 \\ &\forall e \in \mathcal{E}, \forall h \in \mathcal{H}, \forall k \in \mathcal{M} \end{aligned} \quad (35)$$

The maximum demand objective is defined by the following constraints.

$$CP_{\max} \geq \sum_{k \in \mathcal{M}} w_k P_e^k \quad \forall e \in \mathcal{E} \quad (36)$$

The consumption of a batch depends on its duration and on the power of its assigned machine.

$$\begin{aligned} CW_{j,b} &\geq w_k p_{j,b} / v_{j,k} - M(1 - a_{j,b}^k) \\ &\forall (j, b) \in \mathcal{B} \setminus \mathcal{B}^M, \forall k \in \mathcal{M} \end{aligned} \quad (37)$$

The objective function aims at minimizing the total consumption and the maximum demand on peak hours.

$$C \min \alpha \sum_{(j,b) \in \mathcal{B} \setminus \mathcal{B}^M} W_{j,b} + \beta P_{\max} \quad (38)$$

The MILP model made of constraints (12-38) is denoted as (MILP1). We now present an alternative model that contains more binary variables but less constraints. Product variables $x_{j,b}^{e,k}$ and $y_{j,b}^{e,k}$ can be considered as binary and assignment variables $a_{j,b}^k$, as well as start and end variables $x_{j,b}^e$ and $y_{j,b}^e$ can be eliminated by setting:

$$\begin{aligned} a_{j,b}^k &= \sum_{e \in \mathcal{E}} x_{j,b}^{e,k} \quad \forall (j, b) \in \mathcal{B}, \forall k \in \mathcal{M} \\ x_{j,b}^e &= \sum_{k \in \mathcal{M}} x_{j,b}^{e,k} \quad \forall (j, b) \in \mathcal{B}, \forall e \in \mathcal{E} \\ y_{j,b}^e &= \sum_{k \in \mathcal{M}} y_{j,b}^{e,k} \quad \forall (j, b) \in \mathcal{B}, \forall e \in \mathcal{E} \end{aligned}$$

The obtained model is denoted by (MILP2).

If we come back to Figure 1 example. We can use 5 global events with $t_1 = 0, t_2 = 2, t_3 = 3, t_4 = 4$ and $t_5 = 6$. For MILP2, we have $x_{1,1}^{1,1} = y_{1,1}^{2,1} = 1, x_{1,2}^{1,2} = y_{1,2}^{2,2} = 1, x_{2,1}^{4,1} = y_{2,1}^{5,1} = 1, x_{3,1}^{3,2} = y_{3,1}^{5,2} = 1$ and for maintenance operations $x_{4,1}^{2,1} = y_{4,1}^{4,1} = 1$ and $x_{5,1}^{3,2} = y_{5,1}^{5,2} = 1$. For peak computation the only concerned event is $e = 3$ and we have $P_3^2 = 1$ because $P_{3,1} + P_{1,3} + o_{3,2} = 1$ (the event is inside the peak and the machine is on).

5 A simplified model and a matheuristic

The models (MILP1) and (MILP2) proposed in Section 4 have several drawbacks. First they have a large number of binary variables and constraints, especially constraints involving big- M coefficient that significantly weaken the LP relaxation. Second, finding a solution that ends before the common deadline can be difficult and even impossible in practice.

If we ignore the maximum consumption during peak hours, we can simplify the model, transforming the event-based model into a positional model. We consider a set \mathcal{E}_k of maximum $\sum_{j=1}^n B_j$ positions on each machine k and the following reduced set of variables, including a tardiness variable for each job aiming at relaxing the hard common deadline and including it into the objective function.

- $t_{e,k} \geq 0$ is the start time of batch scheduled at position $e \in \mathcal{E}_k$ on machine $k \in \mathcal{M}$.
- $x_{j,b}^{e,k} \in \{0, 1\}$ is equal to 1 if batch $(j, b) \in \mathcal{B}$ is assigned to position e on machine k
- $p_{j,b}^{e,k} \geq 0$ is the production of batch $(j, b) \in \mathcal{B}$ is scheduled at position e on machine k .
- $T_k \geq 0$ is the tardiness of machine $k \in \mathcal{M}$ in the case it finishes its production after the common deadline.

The problem constraints can then be expressed as follows. The start times of tasks at each position are ordered.

$$Ct_{e,k} \leq t_{e+1,k} \quad \forall k \in \mathcal{M}, \forall e \in \mathcal{E}_k \setminus \{|\mathcal{E}_k|\} \quad (39)$$

Each batch (j, b) has to be scheduled at most once.

$$C \sum_{k \in \mathcal{M}_j} \sum_{e \in \mathcal{E}_k} x_{j,b}^{e,k} \leq 1 \quad \forall (j, b) \in \mathcal{B} \quad (40)$$

Each event on each machine (e, k) corresponds to at most one batch.

$$C \sum_{(j,b) \in \mathcal{B}} x_{j,b}^{e,k} \leq 1 \quad \forall k \in \mathcal{M}_j, \forall e \in \mathcal{E}_k \quad (41)$$

The production of a batch (j, b) at a position of a machine is 0 if the batch is not assigned at this position on this machine.

$$Cp_{j,b}^{e,k} \leq D_j x_{j,b}^{e,k} \quad \forall (j, b) \in \mathcal{B}, \forall k \in \mathcal{M}_j, \forall e \in \mathcal{E}_k \quad (42)$$

A batch assigned to a position on a machine must have a minimum size of ϵ .

$$Cp_{j,b}^{e,k} \geq \epsilon x_{j,b}^{e,k} \quad \forall (j, b) \in \mathcal{B}, \forall k \in \mathcal{M}_j, \forall e \in \mathcal{E}_k \quad (43)$$

The demand of each job must be satisfied.

$$C \sum_{b \in \mathcal{B}_j} \sum_{k \in \mathcal{M}_j} \sum_{e \in \mathcal{E}_k} p_{j,b}^{e,k} = D_j \quad \forall j \in \mathcal{J} \quad (44)$$

Setup times must be satisfied between two consecutive batches.

$$Ct_{e+1,k} \geq t_{e,k} + \sum_{(j,b) \in \mathcal{B}} p_{i,b}^{e,k} / v_{i,k} + s_{ij}^k \left(\sum_{b=1}^{B_i} x_{i,b}^{e,k} + \sum_{b'=1}^{B_j} x_{j,b'}^{e+1,k} - 1 \right) \quad \forall e \in \mathcal{E}_k \setminus |\mathcal{E}_k|, \forall i, j \in \mathcal{J}, i \neq j, \forall k \in \mathcal{M} \quad (45)$$

The following constraint computes the violation of the deadline (tardiness) on machine k (in case a job is assigned to the last event, its duration must be added).

$$CT_k \geq t_{|\mathcal{E}_k|} + \sum_{(j,b) \in \mathcal{B}} p_{j,b}^{e,k} - T \quad \forall k \in \mathcal{M} \quad (46)$$

The objective function aims at minimizing a weighted sum of the total tardiness and the total consumption.

$$C \min \gamma \sum_{k \in \mathcal{M}} T_k + \sum_{(j,b) \in \mathcal{B}} \sum_{k \in \mathcal{M}_j} \sum_{e \in \mathcal{E}_k} w_k p_{j,b}^{e,k} / v_{j,k} \quad (47)$$

where γ is a coefficient large enough to ensure that the total tardiness will be minimized in priority.

This model, denoted as (MILP3) is greatly simplified, however the solution ignores peak hours. Consequently in a second phase, The solution of (MILP3) can serve as a basis of a matheuristic for further improvement of the computed peak cost. To that purpose we compute a global event set $\mathcal{E}' = \cup_{k \in \mathcal{M}} \mathcal{E}_k$ and we sort the events according to their time t_e . Then we can accelerate the solution time of (MILP1) or (MILP2) by preassigning the start and end times of activities to the event set according to the solution found by (MILP3).

6 Computational experiments

The experiments are conducted on an Intel Core i7-4770 processor with 4 cores and 8 gigabytes of RAM under the 64-bit Ubuntu 12.04 operating system. We use CPLEX 12.6 with 1 thread for solving the models. The total time limit of 550 seconds for solving both (MILP2) and (MILP3) models: 500 seconds for (MILP3) and 50 seconds for (MILP2).

The instances are extracted from the real instances of [1] and are as follows. For these instances, the number of machines is equal to 3, the time horizon is equal to the number of hours in each month (e.g. in January, this number is $31 \times 24 = 744$) and the machine consumption is constant and equal to $10MW$ for each machine.

Each instance have 10 different lots and the setup time and the speed of each machine is as described by Table 1 and Table 2 respectively. In Table 2, whenever the speed of the machine is equal to zero then the lots can not be scheduled on this machine.

For each instance, the maintenance duration is equal to 120 for each machine. As in Chile, peak hours occurs from 6pm to 11pm, we set the peak hours to these values in our instances.

To solve the problem, we use first the (MILP3) and we use the solution to help the (MILP2) to improve the maximum consumption during peak hours. At

i \ j	1	2	3	4	5	6	7	8	9	10
1	0	4	5	6	7	8	1	1	0	1
2	4	0	4	5	6	7	1	1	0	1
3	5	4	0	4	5	6	1	1	1	1
4	6	5	4	0	4	5	0	2	2	2
5	7	6	5	4	0	4	0	3	3	3
6	8	7	6	5	4	0	0	4	4	4
7	4	4	5	0	0	0	0	5	0	5
8	4	4	5	6	7	8	5	0	0	5
9	4	4	5	6	7	8	0	0	0	3
10	4	4	5	6	7	8	5	5	3	0

Table 1: setup times between batches

lot \ line	1	2	3
1	0	4	0
2	0	5.1	0
3	0	6	0
4	8.1	8.9	0
5	9.5	9	0
6	11	8.8	0
7	10.6	0	4
8	0	0	5.4
9	0	0	8.5
10	0	0	8.1

Table 2: machine speeds for each lot

the beginning of the procedure, i.e. before using the (MILP3), we use a simple heuristic to help the model find a solution. This heuristic considers only one batch for each different lot and schedule this batch on the least loaded machine. This heuristic is described by 1. Maintenance operations are arbitrary set in a way such that, on each machine, half of the batches are scheduled before this operation and the other half are scheduled after. After the resolution of the (MILP3), we use the solution found as a basis solution of the (MILP2).

Experiments have been done on 10 instances and for different numbers of batches, i.e. the number of sublots for each lot: 2, 3 and 4. Results are displayed in Table 3. The first row displayed the percentage of instances solved optimally by (MILP3). The second row describes the percentage of instances solved by both model for each number of batches. The third row presents the average CPU time in seconds needed to solve the instances. The fourth and fifth rows shows the total consumption and the maximum peak of consumption during peak hours.

As we can see, the model solves most of the instances if the number of sublots is set to 2 or 3. However, when 4 sublots are authorized, the number of solved

Algorithm 1 Simple heuristic for finding a simple solution

```
machLoadk = 0,  $\forall k \in \mathcal{M}$   
affj = -1,  $\forall j \in \mathcal{J}$   
for every lot j do  
  minLoad  $\leftarrow \infty$   
  LeastLoadedMach  $\leftarrow -1$   
  for all machine k do  
    if  $k \in \mathcal{M}_j$  and machLoadk < minLoad then  
      minLoad  $\leftarrow \text{machLoad}_k$   
      LeastLoadedMach  $\leftarrow k$   
    end if  
  end for  
  affj = LeastLoadedMach  
  machLoadedk =  $D_j/v_j^k$   
end for
```

	number of sublots		
	2	3	4
% solved opt. by MILP3	50	50	50
% solved (MILP2+MILP3)	37.5	37.5	25
CPU time (s)	37.6	46.8	63.2
Total consumption (MW)	16022.8	16013.9	15892.9
Peak (MW)	30	30	30

Table 3: Results of experiments for (MILP2+MILP3)

instances decreases. This is because of the size of the model which grows with respect to the number of sublots. Another remark can be made about the results of the experiments. Indeed, the peak of consumption is very high. Again, this is mostly due to the large size of the second model.

Table 4 presents the results of experiments on one particular instance. The first and fifth rows show the statut of (MILP3) and (MILP2) respectively. The second and sixth rows present the CPU time needed for solving (MILP3) and (MILP2) respectively and the total CPU time is described in the ninth row. Finally, the total consumption is displayed in the third and seventh rows and the validity of the instance, i.e. the deadline is respected, is described by rows fourth and eighth.

In this table (Table 4), we see that the first model solve the instance optimally but the deadline is not respected for 3 and 4 sublots. The second model does not find an optimal solution but the solution returned is valid, i.e. the deadline is respected, and then the total consumption decreases. Therefore, even if improving the solution of (MILP3) with (MILP2) does not reduce the maximum consumption, this reduces the total consumption and gives a valid solution.

		number of sublots		
		2	3	4
MILP3	Statut MILP3	Optimal	Optimal	Optimal
	CPU time (s) MILP3	0.61	0.91	12.6
	Total consumption 1 (MW)	15186.9	15171.3	15149.8
	Deadline respected	yes	no	no
MILP2	Statut MILP2	Feasible	Feasible	Feasible
	CPU time (s) MILP2	50	50	50
	Total consumption 2 (MW)	15186.9	15161.6	15145.8
	Deadline respected	yes	yes	yes
Total CPU Time (s)		51	52.1	65.7

Table 4: Results of experiments on one instance

7 Conclusion

In this paper, we consider a real industrial problem. For this problem, we present three different models. Both of them solve exactly the problem (MILP1 and MILP2) whereas the last one only minimizes the total consumption (MILP3). To solve the problem, we use first the (MILP3) and we use the solution to help the (MILP1) to improve the maximum consumption during peak hours.

The third model is very efficient as it solved very quickly the model but the second model does not improve the maximum consumption within the time limit allowed. Therefore, an important amount of work is left to be done. There are multiple research direction to pursue. Indeed, an efficient solution method minimizing the maximum consumption during peak hours needs to be developed. For this purpose, several heuristics can be tested in order to improve the quality of the solution in the beginning of the resolution or between the two MILP models.

Another development can be made by improving the modeling of the problem with a Mixed Integer Linear Program in order to decrease the number of constraints and/or variables to simplify the solution procedure. Finally, we aim at designing extended formulations and to develop column generation procedures.

Acknowledgment

This work was funded by ECOS–CONICYT French–Chile cooperation program C13E04.

References

- [1] J. F. Urrutia Jarpa, “Modelo PAP que considera política de control de máxima demanda eléctrica en la producción de empresas manufactureras,” Master’s thesis, Universidad de Concepción, Facultad de Ingeniería, Departamento de Ingeniería Industrial, Chile, 2014.
- [2] R. J. James and B. Almada-Lobo, “Single and parallel machine capacitated lotsizing and scheduling: New iterative mip-based neighborhood search heuristics,” *Computers & Operations Research*, vol. 38, no. 12, pp. 1816–1825, 2011.
- [3] J. Xiao, C. Zhang, L. Zheng, and J. N. Gupta, “Mip-based fix-and-optimize algorithms for the parallel machine capacitated lot-sizing and scheduling problem,” *International Journal of Production Research*, vol. 51, no. 16, pp. 5011–5028, 2013.
- [4] S. Dauzere-Peres and J.-B. Lasserre, “Lot streaming in job-shop scheduling,” *Operations Research*, vol. 45, no. 4, pp. 584–595, 1997.
- [5] J. H. Chang* and H. N. Chiu, “A comprehensive review of lot streaming,” *International Journal of Production Research*, vol. 43, no. 8, pp. 1515–1536, 2005.
- [6] O. Hazir and S. Kedad-Sidhoum, “Batch sizing and just-in-time scheduling with common due date,” *Annals of Operations Research*, vol. 213, no. 1, pp. 187–202, 2014.
- [7] P. Serafini, “Scheduling jobs on several machines with the job splitting property,” *Operations Research*, vol. 44, no. 4, pp. 617–628, 1996.
- [8] W. Xing and J. Zhang, “Parallel machine scheduling with splitting jobs,” *Discrete Applied Mathematics*, vol. 103, no. 1, pp. 259–269, 2000.
- [9] F. Yalaoui and C. Chu, “An efficient heuristic approach for parallel machine scheduling with job splitting and sequence-dependent setup times,” *IIE Transactions*, vol. 35, no. 2, pp. 183–190, 2003.
- [10] D. N. Tahar, F. Yalaoui, C. Chu, and L. Amodéo, “A linear programming approach for identical parallel machine scheduling with job splitting and sequence-dependent setup times,” *International Journal of Production Economics*, vol. 99, no. 1, pp. 63–73, 2006.
- [11] P. Beraldi, G. Ghiani, A. Grieco, and E. Guerriero, “Rolling-horizon and fix-and-relax heuristics for the parallel machine lot-sizing and scheduling problem with sequence-dependent set-up costs,” *Computers & Operations Research*, vol. 35, no. 11, pp. 3644–3656, 2008.
- [12] N. Absi, S. Dauzère-Pérès, S. Kedad-Sidhoum, B. Penz, and C. Rapine, “Lot sizing with carbon emission constraints,” *European Journal of Operational Research*, vol. 227, no. 1, pp. 55–61, 2013.

- [13] G. Mouzon, M. B. Yildirim, and J. Twomey, “Operational methods for minimization of energy consumption of manufacturing equipment,” *International Journal of Production Research*, vol. 45, no. 18-19, pp. 4247–4271, 2007.
- [14] C. Artigues, P. Lopez, and A. Haït, “The energy scheduling problem: Industrial case-study and constraint propagation techniques,” *International Journal of Production Economics*, vol. 143, no. 1, pp. 13–23, 2013.
- [15] G. German, C. Desdouits, and C. Le Pape, “Energy optimization in a manufacturing plant,” in *ROADEF 2015*, Marseille, 2015.
- [16] M. Queyranne and A. Schulz, “Polyhedral approaches to machine scheduling,” Technical University of Berlin, Tech. Rep. 409/1994, 1994.
- [17] G. Riotteau, O. Scaloni, and E. Néron, “Ordonnancement sur des machines identiques avec splitting et temps de préparation dépendant de la séquence,” in *MOSIM '01 : conférence francophone de modélisation et simulation*, 2001.
- [18] F. M. Defersha and M. Chen, “Mathematical model and parallel genetic algorithm for hybrid flexible flowshop lot streaming problem,” *The International Journal of Advanced Manufacturing Technology*, vol. 62, no. 1-4, pp. 249–265, 2012.
- [19] C. A. Floudas and X. Lin, “Mixed integer linear programming in process scheduling: Modeling, algorithms, and applications,” *Annals of Operations Research*, vol. 139, no. 1, pp. 131–162, 2005.
- [20] O. Koné, C. Artigues, P. Lopez, and M. Mongeau, “Comparison of mixed integer linear programming models for the resource-constrained project scheduling problem with consumption and production of resources,” *Flexible Systems and Management Journal*, vol. 25, no. 1-2, pp. 25–47, 2013.
- [21] M. Nattaf, C. Artigues, and P. Lopez, “A hybrid exact method for a scheduling problem with a continuous resource and energy constraints,” *Constraints*, vol. DOI: 10.1007/s10601-015-9192-z, 2015.