Production scheduling of parallel machines with model predictive control

Andrea Cataldo*, Andrea Perizzato, Riccardo Scattolini

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci, 32, 20133 Milan, Italy

Received 27 August 2014
Accepted 12 May 2015

1. Introduction

The energy efficiency of manufacturing production systems is becoming a topic of paramount interest for many reasons, such as the need to minimize the energy consumption of industrial plants, to resize the factory energy supply infrastructures, and to limit the CO_2 emissions, see e.g. European Commission (2009), Organization for Economic Cooperation and Developoment (2004), Seliger (2007), and ICF International (2007). Different levels of manufacturing efficiency have been considered in the literature (Fysikopoulos, Pastras, Alexopoulos, & Chryssolouris, 2014):(i) the process level, which concerns the energy interaction related to the physical machining operations; (ii) the machine level, which considers both processing and auxiliary operations; (iii) the production line level, which refers to a group of different machines and, finally, (iv) the factory level, which concerns the high-level managing of different production lines, possibly interacting and sharing common appliances. In general, improving the efficiency at the lower levels (machine and process) is a complex task because it may result in worsening quality and costs or it may require the deployment of new and more advanced processing techniques. By contrast, energy efficiency at the production or factory level can be improved by designing suitable production scheduling and planning algorithms. This level of optimization is usually preferred because it is less invasive and does not effect quality and costs. For this reason, the development of optimization algorithms for the solution of scheduling problems, such as job shop, flow shop, and flexible flow shop, has been the subject of a huge scientific effort, see e.g. Pinedo (2008) and the references reported there. Recent contributions explicitly dealing with the energy efficient scheduling of production systems are reported in Angel, Bampis, and Kacem (2012), Haït and Artigues (2009), Fang, Uhan, Zhao, and Sutherland (2011), Bruzzone, Anghinolfi, Paolucci, and Tonelli (2012), and Dai, Tang, Giret, Salido, and Li (2013).

This paper considers the problem of optimizing on-line the production scheduling and buffer management of a multiple-line production plant composed by L machines M_i, $i=1,\dots,L$, which can operate at different speeds corresponding to different energy demands. The path from a common source node, where the part to be processed is assumed to be always available, to each machine may differ in the number of buffer nodes, and the energy required to move the part along these transportation lines must be suitably considered in the computation of the overall energy consumption. Therefore, the control problem consists of computing, at each sampling instant, the sequence of commands to be applied to the transportation lines and the processing speed of the machines in order to optimize the throughput of the system and to limit the overall energy consumption. This problem, which shares some similarities with the classical flexible flow shop problem, has been motivated by the optimal management of the de-manufacturing plant described in Colledani, Copani, and Tolio (2014) and Copani et al. (2012). Specifically, this plant is made by a number of machines and a multi-path transportation line, part of which is

* Corresponding author. Tel.: +390223994034.
E-mail addresses: andrea.cataldo@polimi.it (A. Cataldo), andrea.perizzato@gmail.com (A. Perizzato), riccardo.scattolini@polimi.it (R. Scattolini).

made by two parallel independent transport lines which start from the same source node and feed two independent machines. The design of the optimal pallet routing has been already considered in Cataldo and Scattolini (2014a, 2014b), where however the target machine of each pallet has been assumed to be a priori given.

The optimal scheduling of parallel machines, which must guarantee the completion of a given number of tasks by assigning them to different machines, has been considered in many papers, see e.g. the review (Senthilkumar & Narayanan, 2010) and the references therein. This problem is known to be very complex, see Weng, Lu, and Ren (2001), and therefore the proposed solutions are mainly based on the development of heuristics, see e.g. Kim, Na, and Chen (2003), Rabadi, Moraga, and Al-Salem (2008), and Jansen and Porkolab (2001). On the contrary, the approach here proposed relies on Model Predictive Control, or MPC, see e.g. Camacho and Bordons Alba (2004), a technique widely popular in the process industry, such as chemical, petrochemical, pulp and paper, but still in its infancy in the field of discrete manufacturing, save for the notable exceptions of Vargas-Villamir and Rivera (2001), Vargas-Villamir and Rivera (2000), Alessandri, Gaggero, and Tonelli (2011), Braun, Rivera, Flores, Carlyle, and Kempf (2003), Ydstie (2004), Ferrio and Wassick (2008), Wang and Rivera (2008), where problems related to the management of supply chains have been studied. MPC is based on the simple idea of transforming the control problem into an optimization one, where different goals and constraints on the process variables can be included. Specifically, in the problem here considered, MPC recursively computes the optimal sequence of buffer commands and machines' processing times over a given prediction horizon by minimizing the overall energy consumption and maximizing the future production. Then, only the first value of the optimal sequence is applied and the overall optimization procedure is repeated at the next time instant. Optimization is performed under suitable constraints on the production, on the electric power involved, and under the physical constraints imposed by the system. These constraints are described by logical statements, which in turn are transformed into algebraic relations among boolean variables, see Bemporad and Morari (1999). Moreover, the machines are represented by finite state machine models, so that the overall system to be optimized is described by a Mixed Logical Dynamic (MLD) model, see again Bemporad and Morari (1999), a representation which combines discrete time dynamics with logical (boolean) decision variables. Thanks to the use of MLD models, the resulting optimization problem belongs to the class of Mixed Integer Linear Programming (MILP) problems, for which fast solvers are available.

The paper is organized as follows. In Section 2 the problem is stated, the models of the components, i.e. machines and buffer zones, are developed, and the overall MLD model is derived. In Section 3 the optimization problem is formulated and its main characteristics are examined. Section 4 is devoted to present and critically analyze some simulation results where different scenarios are considered. Finally, in Section 5 some conclusions and hints for future developments close the paper.

2. Problem formulation

The generic structure of the production system considered in this paper is sketched in Fig. 1: it consists of L parallel production lines, each one with p_i, $i=1,\dots,L$ buffer nodes ended by a machine M_i (Fig. 2). The machines $M_1,\dots,M_L$ are assumed to have a controllable and variable duration processing time related to the required energy to perform the machining operations. This means that it is possible to choose whether a machine must process the next part at full or slow speed with consequent high or low energy demand. The binary variable $u_{i,j}$, $i=1,\dots,L, j=1,\dots,p_i$ represents the trigger which moves the part from node $N_{i,j-1}$ to $N_{i,j}$. Specifically, $u_{i,j}=0$ if the part is not moved from $N_{i,j-1}$ to $N_{i,j}$ or if node $N_{i,j-1}$ is empty, while $u_{i,j}=1$ if the part is moved from $N_{i,j-1}$ to $N_{i,j}$.

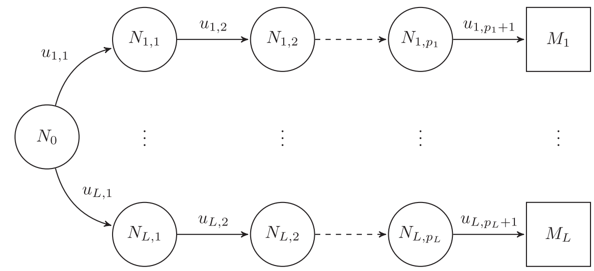

Fig. 1. Production system.

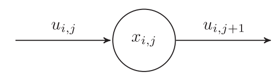

Fig. 2. Node model.

The control problem consists in moving the parts from the root node N_0 to the machines $M_1,\dots,M_L$ and in deciding the processing time of each machine, while ensuring that constraints on maximum power and minimum production are fulfilled.

Assumption 1. The root node N_0 always contains a part, i.e. there is always a part ready to be processed by the system.

2.1. Node model

Let $x_{i,j}$ be a logical state related to node $N_{i,j}$ and let $x_{i,j}=1$ when $N_{i,j}$ contains a part and $x_{i,j}=0$ otherwise. The variable $u_{i,j}$ is logical as well, and $u_{i,j}=1$ means that the part will be moved from $N_{i,j-1}$ to $N_{i,j}$.

Letting k be the discrete time index, the dynamics of the logic state is given by

$$x_{i,j}(k+1)=x_{i,j}(k)+u_{i,j}(k)-u_{i,j+1}(k) \tag{1}$$

In order to simplify the notation, from now on we will drop the time index k when not required for clarity of presentation. Moreover, the superscript $^+$ will denote the variable at the next time instance, so that, given a generic variable $\varphi(k)$, the symbol φ will correspond to $\varphi(k)$ and φ^+ to $\varphi(k+1)$. According to this notation, (1) can be written as

$$x_{i,j}^+=x_{i,j}+u_{i,j}-u_{i,j+1} \tag{2}$$

The inputs $u_{i,j}$ must be suitably constrained in order to prevent the states taking values different from zero and one, and to avoid unrealistic configurations, such as moving a part out of an empty node. In particular, it is possible to move a part into $N_{i,j}$ if and only if all the following conditions are fulfilled:

1. The node $N_{i,j-1}$ contains a part .
2. The node $N_{i,j}$ is empty or it contains a part which is moved to $N_{i,j+1}$ at the same time instant.

These conditions can be rewritten using logical operators as

$$\neg x_{i,j-1} \wedge (\neg x_{i,j} \vee (x_{i,j} \wedge u_{i,j+1})) \tag{3}$$

which, according to the propositional calculus rules (Bemporad & Morari, 1999; Lucas, Mitra, & Moody, 1992; McKinnon & Williams, 1989; Raman & Grossmann, 1991; Williams, 2013), is equivalent to

$$u_{i,j} \le x_{i,j-1}$$
$$u_{i,j} \le 1-x_{i,j}+u_{i,j+1} \tag{4}$$

As for the root node N_0, Assumption 1 implies that $x_0 = 1$ at any time instant, allowing its dynamics to be neglected. However, only one part at a time can be moved out of N_0 and thus the following constraint must be fulfilled:

$$\sum_{i=1}^{L} u_{i,1} \leq 1 \tag{5}$$

The interface between the final node N_{i,p_i} of each line and the corresponding machine M_i will be discussed in the following together with the models of the machines.

2.2. Simplified Machine Model (SMM)

Let the i-th machine M_i, $i = 1, \ldots, L$, be represented by a finite state machine which describes the following behavior:

- the machine can either be *busy* (B) or *free* (F);
- the transition $F \to B$ occurs on the rising edge of a logic (binary) input, denoted by γ_i, which may be seen as a "starting" signal;
- together with the starting signal, a second integer input, denoted with η_i, must be set. The machine will then stay in the busy state for η_i time instants and, afterwards, it will go back to be free, i.e. the transition $B \to F$ will occur;
- when the machine is busy, the output representing the absorbed power, denoted by q_i, must be set according to a properly designed function $q_i = f_{e_i}(\eta_i)$, related to the processing time;
- during the last time instant in which the machine is busy, the logic output representing the end of the processing, denoted by ψ_i, must be set;
- when busy, the logic output β_i, representing the busy state of the machine, must be set to one.

Therefore, each machine M_i can be represented by the block shown in Fig. 3.

2.2.1. Mathematical model

In view of the previous definitions, the mathematical model of M_i can be derived by defining the following internal states:

- z_i^l: logic state denoting whether the machine is free ($z_i^l = 0$) or busy ($z_i^l = 1$);
- z_i^c: integer state counting the number of time instants in the busy state;
- z_i^η: integer state to "hold" the value of η_i during the transition $F \to B$.

In addition, consider the following auxiliary logical variable:

$$\mu_i = 1 \leftrightarrow z_i^c \geq z_i^\eta - 1 \tag{6a}$$

which will be used to trigger the transition $B \to F$, as illustrated in Fig. 4.

The dynamics of the states is given by

$$z_i^{l+} = (z_i^l \wedge \neg\mu_i) \vee (\neg z_i^l \wedge \gamma_i) \tag{6b}$$

$$z_i^{c+} = \begin{cases} z_i^c + 1 & \text{if } z_i^l \\ 0 & \text{if } \neg z_i^l \end{cases} \tag{6c}$$

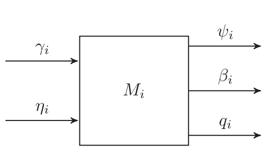

Fig. 3. Machine block.

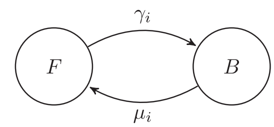

Fig. 4. FSM of a machine.

$$z_i^{\eta+} = \begin{cases} z_i^\eta & \text{if } z_i^l \\ \eta_i & \text{if } \neg z_i^l \end{cases} \tag{6d}$$

These relations describe the following facts:

- (6b) states that the machine will be busy at the next time instant either if it is currently busy and the processing is not yet completed, i.e. $\mu_i = 0$, or if it is currently free and the starting trigger γ_i is set;
- (6c) defines z_i^c as an increasing counter only when busy, which is reset when free;
- (6d) defines z_i^η as a holder of the last value that η_i has taken when in the free state. This value is kept as long as the machine is busy.

The absorbed power is defined as a function of the value of z_i^η, since it is different from zero only when the machine is in the busy state and is related to the production duration. No assumptions on the class of this function have been made,[1] but since $z_i^\eta \in \mathbb{N}$, it is required to be defined only for integer positive values. In particular, assume that $\eta_i \in [\underline{\eta}_i, \overline{\eta}_i] \subseteq \mathbb{N}$, $\overline{\eta}_i \geq \underline{\eta}_i \geq 1$ and let $D_i = \overline{\eta}_i - \underline{\eta}_i + 1$ be the number of possible values η_i can take. The function $q_i = f_e(z_i^\eta)$ can then be obtained by defining D_i auxiliary logical variables, denoted by δ_i^h, $h = 1, \ldots, D_i$ constrained as follows:

$$\sum_{h=1}^{D_i} \delta_i^h = 1 \tag{6e}$$

$$\sum_{h=1}^{D_i} h\delta_i^h = z_i^\eta \tag{6f}$$

The absorbed power is then computed as

$$q_i = \begin{cases} \displaystyle\sum_{h=1}^{D_i} f_{e_i}(h)\delta_i^h & \text{if } z_i^l \\ 0 & \text{if } \neg z_i^l \end{cases} \tag{6g}$$

The other two logic outputs can be easily obtained from the state variables as

$$\beta_i = z_i^l \tag{6h}$$

and, for the end of cycle

$$\psi_i = z_i^l \wedge \mu_i \tag{6i}$$

which can be recast as a set of inequalities as described in Bemporad and Morari (1999).

2.2.2. Machine–buffer interface

The machines must be properly connected to the buffer lines as in Fig. 1. It can be noted that each machine behaves similarly to a node. The main difference is in the transition from being busy to free. In fact, each buffer node $N_{i,j}$ can be freed by setting the controllable variable $u_{i,j+1}$, while the machines are automatically freed once the processing is completed. Therefore, with respect to

[1] It should be noted that the function $q_i = f_{e_i}(\eta_i)$ should be monotonically decreasing, because, in general, the faster processing (i.e. small η_i), the higher absorbed power (i.e. big q_i). However, at least from a theoretical point of view, nothing prevents it to behave differently.

Fig. 1 and the notation presented in Section 2.1, each machine can be seen as a node by setting

$$x_{i,p_i+1} = \beta_i$$
$$u_{i,p_i+1} = \gamma_i \tag{7}$$

Concerning constraints (4), due to the inability to pull a part out of the machine, a part in N_{i,p_i+1} can be moved into M_i only when M_i is free, i.e. $x_{i,p_i+1} = 0$. Therefore, the set of constraints for the last node of each line is

$$u_{i,p_i+1} \leq x_{i,p_i}$$
$$u_{i,p_i+1} \leq 1 - x_{i,p_i+1} \tag{8}$$

The underlying assumption is that each machine must be free for at least one time instant before a new part can be processed. This assumption will be removed in the next paragraph in which an Enhanced Machine Model is introduced.

2.3. Enhanced Machine Model (EMM)

The model presented in Section 2.2 is valid only when the machine cannot achieve continuous processing operations, meaning that it is not possible to unload the machined part and load a new one at the same time. In fact, the previous model forces the machine to be free for at least one time instant (unload operation) before going back to busy. However, some industrial applications may achieve continuous processing operations and therefore an Enhanced Machine Model is developed to take this behavior into account.

As for the simpler model, consider the state machine in Fig. 5:

- *Free* (F): the machine is not operating and waiting for a part to be processed.
- *Busy* (B): the machine is operating and it is not the last operative time instant.
- *End* (E): the machine is operating in its last operative time instant.

The main difference with the previous model is the ability to directly switch from the end state to the busy one, i.e. to begin the machining of a new part immediately after the current one has finished. There is no need to go through the free state. Inputs and outputs are defined in the same way as in the previous model. However, the output power q_i must be different from zero when the active state is either busy (B) or end (E). In fact, these states only differ from a timing point of view, i.e. E is active during the last time instant of the processing, while B during the previous ones.

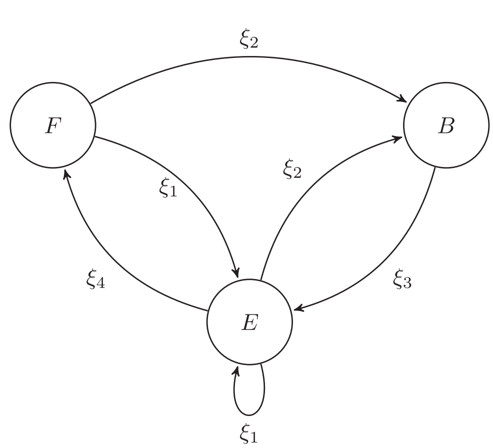

Fig. 5. FSM of a enhanced machine.

2.4. Mathematical model

In the following, to simplify the notation the index i denoting the specific machine will be dropped. Consider the following state variables:

- z^F: logic state which is true when the machine is free.
- z^B: logic state which is true when the machine is busy.
- z^E: logic state which is true when the machine is in the end state.
- z^c: integer state counting the time instants spent on the busy state.
- z_η: integer state "holding" the value of η set during the transition $F \rightarrow B$ and $F \rightarrow E$.

The transitions are defined by the following auxiliary logical variables:

$$\xi_1 = 1 \leftrightarrow \gamma \wedge \eta = 1 \tag{9a}$$

$$\xi_2 = 1 \leftrightarrow \gamma \wedge \eta > 1 \tag{9b}$$

$$\xi_3 = 1 \leftrightarrow z^c \geq z^\eta - 2 \tag{9c}$$

$$\xi_4 = 1 \leftrightarrow \neg\gamma \tag{9d}$$

In other words, the following scenarios are possible:

1. From the free state, it is possible to go to the busy or end states depending on the value of the processing time η. In particular, if $\eta > 1$ the active state will be B and a long processing will start, otherwise it will be E leading to a new one-step machining.
2. When the active state is B, the machine will remain busy until the remaining processing time is only one step, that is $\xi_3 = 1$. At this point, the machine will switch to E.
3. From the E state it is possible to:
   (a) Begin a new long processing ($\eta > 1$) by switching to B.
   (b) Begin a new one-step processing ($\eta = 1$) by staying in E.
   (c) End the current processing by going back to F.

The dynamics of the state is given by

$$z^{F+} = (z^F \wedge \neg\xi_2) \vee (z^E \wedge \xi_4) \tag{9e}$$

$$z^{B+} = (z^F \wedge \xi_2) \vee (z^B \wedge \neg\xi_3) \vee (z^E \wedge \xi_2) \tag{9f}$$

$$z^{E+} = (z^F \wedge \xi_1) \vee (z^B \wedge \xi_3) \vee (z^E \wedge \xi_1) \tag{9g}$$

$$z^{c+} = \begin{cases} z^c + 1 & \text{if } z^B \\ 0 & \text{if } \neg z^B \end{cases} \tag{9h}$$

$$z^{\eta+} = \begin{cases} z_i^\eta & \text{if } z^B \\ \eta_i & \textit{otherwise} \end{cases} \tag{9i}$$

Concerning the outputs, (6e) and (6f) still apply in this context. However, (6g) must be changed to

$$q = \begin{cases} \sum_{h=1}^{D} f_e(h)\delta^h & \text{if } z^B \vee z^E \\ 0 & \textit{otherwise} \end{cases} \tag{9j}$$

The busy output β is now defined as

$$\beta = z^B \vee z^E \tag{9k}$$

and the end of cycle ψ

$$\psi = z^E \tag{9l}$$

The logical states z^B, z^F and z^E must also be constrained as follows:

$$z^F + z^B + z^E = 1 \tag{9m}$$

In fact, although the combination of the graph topology in Fig. 5, together with the definitions of ξ_1, ξ_2, ξ_3 and ξ_4 and the dynamics of z^B, z^F and z^E prevent the activation of more that one state at a time, constraint (9m) is still required to avoid the potential feasibility of the initial state $z^F = z^B = z^E = 0$, which would lead to a wrong evolution of the system.

As in the case of the Simplified Machine Model, model (2.4) belongs to the class of Mixed Logical Dynamics models (MLD) and thus it can be reformulated as a suitably constrained linear system (Bemporad & Morari, 1999).

2.4.1. Machine–buffer interface

The machine–buffer interface is very similar to what discussed in Section 2.2.2. In particular, (7) still applies for the enhanced model with β defined in (9k). By contrast, it is now possible to start the machining of a new part when the part being currently processed is about to end. More specifically, if the active state of M_i is E, which corresponds to $\psi_i = 1$, the machine can process a new part. Such behavior is achieved by setting the constraints in place of (8)

$$\begin{aligned}&u_{i,p_i+1} \le x_{i,p_i}\\&u_{i,p_i+1} \le 1 - x_{i,p_i+1} + \psi_i\end{aligned} \tag{10}$$

From a conceptual point of view, it can be noted that the role of ψ_i in (10) is equal to that of $u_{i,j+1}$ in (4). Indeed, ψ_i can be seen as a signal which moves the part out of the machine M_i in the same way as $u_{i,j+1}$ moves it out of $N_{i,j}$.

2.5. Mixed Logical Dynamical model of the system

The system described in the previous sections is characterized by the mutual coexistence of discrete time dynamics and logical variables. Such systems are referred as *hybrid*. Different classes of hybrid dynamical models have been developed in the literature: Piecewise Affine (PWA), Mixed Logical Dynamical (MLD), Linear Complementary (LC), Extended Linear Complementary (ELC) and Max–Min-Plus-Scaling (MMPS) Systems (see Heemels, De Schutter, & Bemporad, 2001 and the references therein). From now on, MLD models (Bemporad & Morari, 1999) will be considered since they are particularly suitable for control purposes.

As discussed in Bemporad and Morari (1999) a MLD system can be described by the following linear relations:

$$\begin{cases} x(k+1) = Ax(k) + B_u u(k) + B_\delta \delta(k) + B_z z(k) \\ y(k) = Cx(k) + D_u u(k) + D_\delta \delta(k) + D_z z(k) \\ E_\delta \delta(k) + E_z z(k) \le E_u u(k) + E_x x(k) + E \end{cases} \tag{11}$$

where

- $x = [x_c, x_l]$ is the state vector and $x_c \in \mathbb{R}^{n_c}$, $x_l \in \{0,1\}^{n_l}$.
- $u = [u_c, u_l]$ is the input vector and $u_c \in \mathbb{R}^{m_c}$, $u_l \in \{0,1\}^{m_l}$.
- $y = [y_c, y_l]$ is the output vector and $y_c \in \mathbb{R}^{p_c}$, $y_l \in \{0,1\}^{p_l}$.
- $\delta \in \{0,1\}^{r_l}$ is a vector of binary auxiliary variables.
- $z \in \mathbb{R}^{r_c}$ is a vector of continuous variables.

In Bemporad and Morari (1999) it is shown how logical relationships, such as (6b), as well as implications between continuous and binary variables, e.g. (6c), can be recast as a set of linear inequalities by introducing a certain number of auxiliary variables, resulting in a model in the form of (11).[2]

[2] For the Simplified Machine Model: $n_c = 7$, $n_l = 2$, $m_c = 2$, $m_l = 5$, $p_c = 2$, $p_l = 4$, $r_c = 6$, $r_l = 10$. For the Enhanced Machine Model: $n_c = 7$, $n_l = 6$, $m_c = 2$, $m_l = 5$, $p_c = 2$, $p_l = 4$, $r_c = 6$, $r_l = 24$.

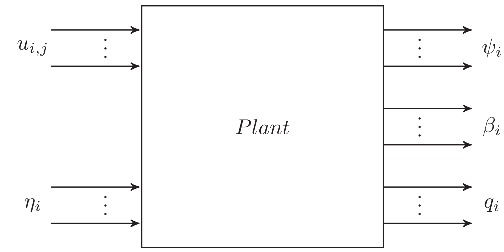

Fig. 6. Schematic block of the plant.

2.6. Overall plant model

In summary, the overall system to be controlled is defined by the connection of all the nodes and the machines accordingly to the plant topology. Therefore, the plant model can be seen as the block shown in Fig. 6 where:

- $u_{i,j}$, $i = 1, \dots, L$, $j = 1, \dots, p_i$ are controllable inputs which cause the parts to move and the machines to start;
- η_i, $i = 1, \dots, L$, are the controllable inputs which define the processing time of each part;
- q_i, ψ_i and β_i, $i = 1, \dots, L$, are the measured outputs as described in Section 2.2.

Note that the use of either the simple of the Enhanced Machine Model is transparent with respect to the definition of inputs and outputs, since they differ only in the internal dynamics. Therefore, it is possible to define a unique control problem which will adapt itself to the actual model of the machines being used. Moreover, it is also possible to include configurations in which both models coexist to describe the behavior of different machines.

3. Model predictive control

The MPC problem consists of maximizing the production while fulfilling constraints on the overall energy consumption and minimum production. The resulting constrained optimization problem must be solved on-line by computing the predicted evolution of the system behavior as a function of the available control variables in a future time window called *prediction horizon*. The output of the optimization process is the optimal sequence of inputs in the horizon. Then, accordingly to the so-called *Receding Horizon*, or *Rolling Horizon*, (RH) approach, only the first optimal input is applied to the system and the others are discarded. The optimization process is then repeated at the next time instant. The cost function to be minimized includes the following terms:

1. *Production*: a negative cost in order to maximize the production. It can be measured in terms of end-of-cycles of the machines, indicated by the outputs ψ_i, $i = 1, \dots, L$;
2. *Energy consumption*: a positive cost to minimize the overall energy consumption;
3. *Movements*: a positive term weighting useless movements of the parts;
4. *Part*: a positive weight penalizing the presence of parts in the nodes if not needed to fulfil the requirements.

Based on the previous considerations, the cost function has been selected as follows:

$$\begin{aligned}J_1 = &-Q_{prod} \sum_{k=t}^{t+N-1} \sum_{i=1}^{L} \psi_i(k) + Q_{energy} \Delta t \sum_{k=t}^{t+N-1} \sum_{i=1}^{L} q_i(k)\\ &R_{move} \sum_{k=t}^{t+N-1} \sum_{i=1}^{L} \sum_{j=1}^{p_i} u_{i,j}(k) + Q_{part} \sum_{k=t}^{t+N-1} \sum_{i=1}^{L} \sum_{j=1}^{p_i} x_{i,j}(k)\end{aligned} \tag{12}$$

where N is the prediction horizon, t is the current time index, Δt is the adopted sampling time and the weights Q_{prod}, Q_{energy}, R_{move} and Q_{part} are design parameters. The maximum absorbed power and minimum production requirements along the prediction horizon can be included in the optimization problem by defining q_{max} as the overall maximum allowable power and P_{min} as the minimum allowable production. Then, the following constraints must be considered:

$$\sum_{i=1}^{L} q_i(k) \le q_{max} \tag{13a}$$

$$\sum_{k=t}^{t+N-1} \sum_{i=1}^{L} \psi_i(k) \ge P_{min} \tag{13b}$$

In order to avoid infeasibility problems, due to the potential impossibility to contemporarily fulfill the above relations, constraints (13) can be modified to allow their violation when necessary by adding slack variables ε_q and ε_p as follows:

$$\sum_{i=1}^{L} q_i(k) \le q_{max} + \varepsilon_q \tag{14a}$$

$$\sum_{k=t}^{t+N-1} \sum_{i=1}^{L} \psi_i(k) \ge P_{min} - \varepsilon_p \tag{14b}$$

$$\varepsilon_q, \varepsilon_p \ge 0 \tag{14c}$$

These slack variables must be heavily weighted in the cost function in order to prevent them to be different from zero when the optimization problem is feasible, so that the cost function becomes

$$J_2 = J_1 + S_p \varepsilon_p + S_q \varepsilon_q \tag{15}$$

where the coefficients S_p and S_q take sufficiently high values. Therefore the optimization problem can be stated as

$$\min_{u_{i,j},\, \eta_i,\, i=1,\dots,L, j=1,\dots,p_i, \varepsilon_p, \varepsilon_q} J_2 \tag{16}$$

subject to (14).

3.1. Avoiding deadlocks and guaranteeing due date

The previous formulation of the MPC optimization problem does not guarantee a given due date, a property often required in industrial applications. In fact, the constraints (14) are implemented according to the RH approach, and the optimal solution can even cause deadlocks of the system when the minimum production requirement is small. This is shown in the very simple scenario depicted Fig. 7, where it is assumed that the prediction horizon is $N=5$ steps, the minimum production requirement is $P_{min}=1$ and the maximum processing time is $\eta=2$. The production requirement can be satisfied by operating at the slower speed (i.e. $\eta=2$), and the controller will choose this scenario in order to minimize the absorbed power, given that $f_e(1) > f_e(2)$. However, two solutions still solve the problem; in fact, since only four time steps are required to process the part and the horizon is one step longer, the remaining step can be put on the leading (Fig. 8a) or trailing (Fig. 8b) edge of the horizon. These solutions have the same optimal cost and thus the controller will randomly choose one of them. However, the first one (Fig. 8a) must be avoided, since it locks the system due to the RH implementation. In the

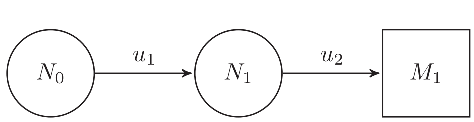

Fig. 7. Example of a simple configuration.

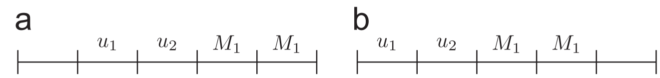

Fig. 8. Two feasible solutions for the same problem. (a) First solution: initial delay. (b) Second solution: no initial delay.

following, two solutions to this problem are proposed. The first one does not introduce hard constraints, but does not guarantee a given level of production, while the second one is characterized by additional constraints which enforce the fulfillment of a prescribed due date.

3.1.1. Exponential weighting

This solution consists in increasingly weighting the control variables along the horizon, so that the optimal solution tends to activate the control variables at the beginning of the prediction horizon. With reference to the previous example, this means that the former solution is "cheaper" than the latter. The new cost function is

$$J_3 = J_2 + \sum_{k=t}^{t+N-1} R_{dead}^{k-t} \sum_{i=1}^{L} \sum_{j=1}^{p_i} u_{i,j}(k) \tag{17}$$

where $R_{dead}^0 < R_{dead}^1 < \dots < R_{dead}^{N-1}$. In addition, R_{dead}^{N-1} must be much smaller than any other weight appearing in (15) in order for the optimal solution to be unaffected, besides the removal of any initial delay.

3.1.2. Guaranteed due date

In the second solution, the production P_{min} is guaranteed at every N time steps. This can be achieved by letting $\bar{t} = \lfloor t/N \rfloor N$ and by introducing in the optimization problem the additional constraint

$$\sum_{k=\bar{t}}^{\bar{t}+N-1} \sum_{i=1}^{L} \psi_i(k) \ge P_{min} \tag{18}$$

which is not forced in a RH form, but makes reference to successive time intervals of length N. Also in the case of these additional constraints, the use of suitable slack variables can be necessary to avoid feasibility properties, as discussed in (14) and (15) for constraints (13). In any case, the resulting optimization problem is more stringent than the one with the exponential weighting, so that the feasibility issue can become more critical.

3.1.3. Storage costs

The parts produced in the N time steps considered the due date constraint have usually to be stored before their delivery, with related storage costs. In order to minimize these costs, the production can be weighted in the performance index so as to postpone it as long as possible. This can be easily considered in the problem formulation with the new performance index

$$J_4 = J_3 + \sum_{k=\bar{t}}^{\bar{t}+N-1} \left[Q_{store} \sum_{i=1}^{L} \psi_i(k) \right] \tag{19}$$

where Q_{store} is a proper weight.

4. Simulation experiments

The algorithm here proposed has been implemented in MATLAB using the MPT Toolbox (Herceg, Kvasnica, Jones, & Morari, 2013) and the HYSDEL (Torrisi & Bemporad, 2004) modeling language. In the following, two experiments are described to highlight its main features. The first one focuses on the analysis of the system behavior in response to variations of the minimum production regardless of the energy consumption, while the second one on the production maximization constrained by a limited amount of available power. Both experiments are based on the plant depicted in Fig. 9, where the two machines can process the parts at slow speed (two time instants, $\eta = 2$) or at full speed (one time instant, $\eta = 1$), but they differ in the absorbed power, as listed in Table 1.

Note that the absorbed power refers to the whole processing which is assumed to be uniformly split during the machining. For instance, M_2 requires an overall energy of $2.00\Delta t$ (kJ), where Δt (s) is the adopted sampling time measured in seconds, to process a part at slow speed, that is in two time instants. Some remarks about the selected values are in order:

- as expected, slow processing times lead to lower absorbed power;
- M_1 is always more powerful than M_2, given the same processing time. However, from Fig. 9, it can be noted that M_2 is farther than M_1 from the source node. Therefore, M_2 may not always be the obvious optimal choice.

The following tuning parameters are used in both experiments:

- The prediction horizon is N=6.
- The sampling time is $\Delta t = 60$ (s).
- The movement cost is $R_{move} = 0$.
- The storage cost is $Q_{store} = 0$.
- The weights of the slack variables are $S_q = 10^6$ and $S_p = 10^4$.

Unless otherwise specified, the exponential weighting method described in Section 3.1.1 to avoid deadlocks has been used with $R^h_{dead} = 0.01(h+1)$, $h = 0, \ldots, N-1$. Each test has been run with the two machine models *SMM* and *EMM* to evaluate the different behaviors.

4.1. *Minimum production*

The aim of this experiment is to demonstrate the scheduling capabilities of the algorithm in case of dynamic variations of the

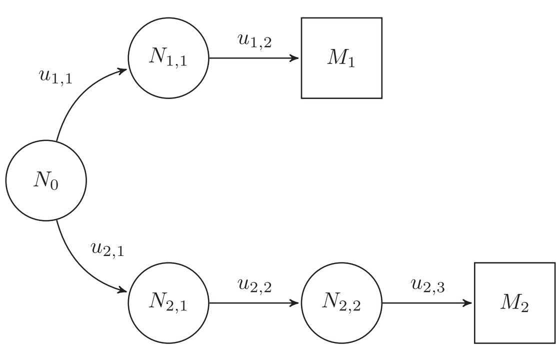

Fig. 9. Experimental plant.

Table 1
Machines absorbed power.

Machine	Processing time instants	Absorbed power (kW)
M_1	1	2.40
M_1	2	1.05
M_2	1	2.20
M_2	2	1.00

minimum production requirement. The cost function parameters have been chosen as $Q_{part} = 10$, $Q_{prod} = 1.2 \times 10^5$, $Q_{energy} = 1$. Note that, even if $Q_{prod} \gg Q_{energy}$, the energy consumption has a more relevant impact on the performance index since the power is measured in terms of [W] and the energy magnitude is 10^5, then it is expected to be minimized with the production as low as required to fulfil the minimum specification. Moreover, if the minimum production is set to zero, the trivial solution of turning all the plant off will be the optimal one to minimize the energy consumption.

In the simulation experiment, the maximum power constraint (13) has been neglected and the minimum production has been changed as follows:

- at t=0 [min] only one part must be processed during the prediction horizon, i.e. it has been set $P_{min} = 1$;
- at t=20 (min), $P_{min} = 2$;
- at t=40 (min), $P_{min} = 4$;
- at t=60 (min), $P_{min} = 6$;
- at t=80 (min), $P_{min} = 8$.

4.1.1. *Simplified Machine Model*

Fig. 10 shows the resulting scheduling of the machines as well as the buffer management when the model described in Section 2.2 is used. The dashed vertical lines in Fig. 10a denote the changes in the minimum production requirement. The following remarks can be stated:

- from t=0 (min) to t=20 (min), machine M_2 is preferred to M_1. This is not surprising since the low production allows the controller to choose the cheapest machine at slow speed, that is M_2 for two time steps;
- from t=20 (min) to t=40 (min), machine M_2 is still preferred to M_1. In order to fulfil the higher production requirement, the idle time of M_2 is reduced;
- from t=40 (min) to t=60 (min), M_1 is periodically scheduled at slow speed;
- from t=60 (min) to t=80 (min), the production requirement further increases and both machines switch at full speed, as expected;
- from t=80 (min) to t=100 (min), both machines already operate at full speed but the production is further increased. This results in a missing production, as shown in Fig. 10b.

The previous experiment has been repeated by substituting the exponential weighting with the due date constraint described in Section 3.1.2. The results achieved are reported in Fig. 11. A direct comparison between Figs. 10 and 11 show that the exponential weighting method produces a slightly more regular solution in terms of machines use and energy consumption. Moreover, if the minimum production requirement changes during the fixed time window where the constraint (18) is active, the due date constraint can lead to feasibility problems, see time t=40 (min) for example. For this reason, the exponential weighting can be preferred when there are not hard production constraints.

The same experiment has been performed by including in the performance index to be minimized a term weighting the storage time, as shown in (19). Specifically, the weight $Q_{store} = 3 \times 10^4$ has been used and the results obtained are shown in Fig. 12. By comparing the results of Figs. 11 and 12, it is easy to see the different behavior of the machines in the interval from $t = 20$ (min) to $t = 40$ (min), in which the machines use a higher power and then work faster in order to reduce the storage time of the produced parts.

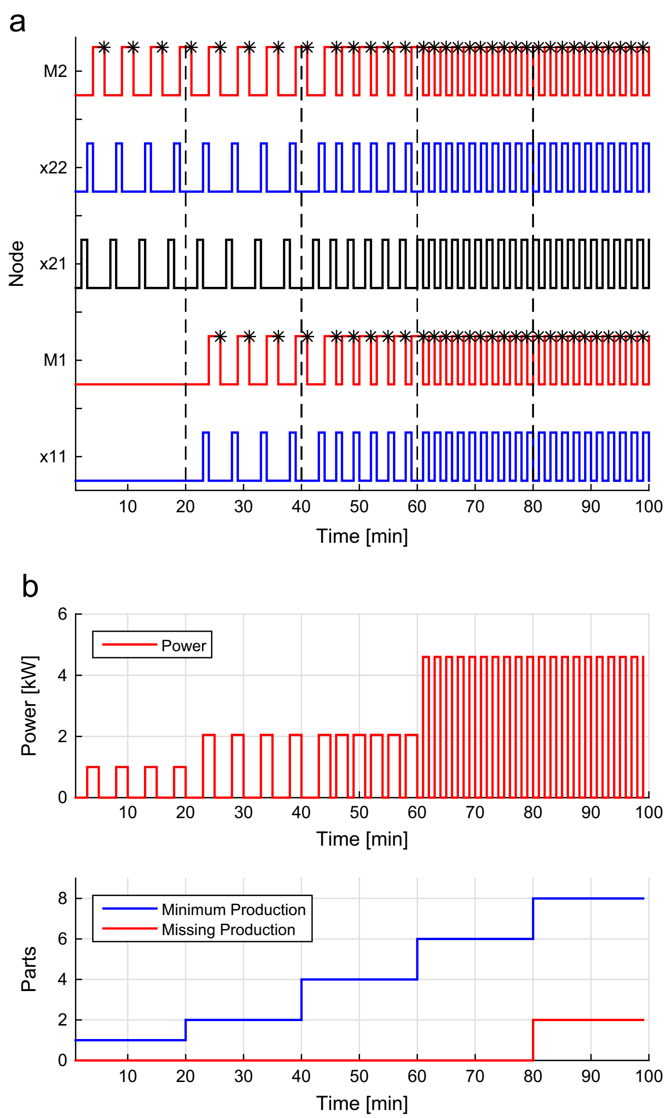

Fig. 10. Power Minimization using the simple machine model. (a) Machine scheduling and buffer management. The black asterisks denote the end of the machining. (b) Production requirement, missing production (bottom) and absorbed power (top).

4.1.2. *Enhanced Machine Model*

The plots related to the use of the Enhanced Machine Model are illustrated in Fig. 13. The following observations can be made:

- from t=0 (min) to t=20 (min), the optimal scheduling is equal to the previous case. In fact, the minimum production is low enough to avoid any continuous machining;
- from t=20 (min) to t=40 (min), M_1 is never turned on. On the other hand, M_2 takes advantage of the continuous machining to fulfil the requirements. Note that this solution is comparable to the previous case. Indeed, the only difference is in the different scheduling of the idle time of M_2;
- from t=40 (min) to t=60 (min), M_2 is continuously processing the parts at slow speed, while M_1 periodically operates at slow speed. Comparing these results to those of Fig. 10, the idle time of the more expensive M_1 is now bigger, leading to a lower energy consumption;
- from t=60 (min) to t=80 (min), both machines are continuously scheduled at slow speed. Note that in Fig. 10 they were periodically scheduled at full speed, but the same production is achieved;

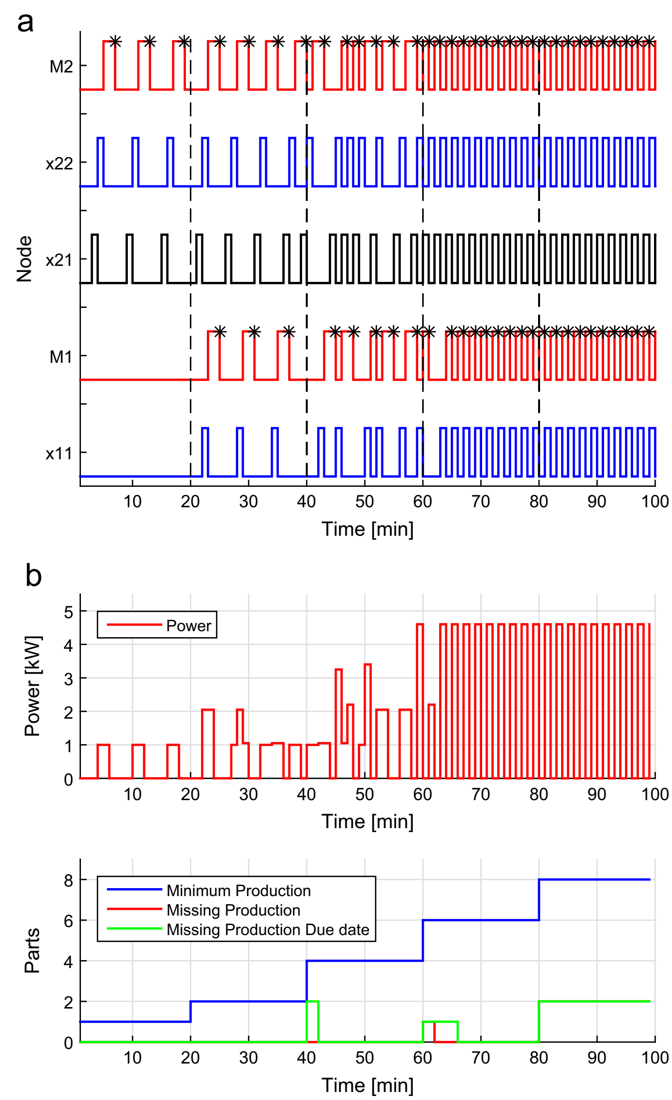

Fig. 11. Minimum production constrained problem with Due date constraint. (a) Machine scheduling and buffer management. The black asterisks denote the end of the machining. (b) Production and absorbed power.

- from t=80 (min) to t=100 (min), a missing production occurs, as depicted in Fig. 13b. It may be argued that the machines should both operate at full speed, which is not possible due to the plant topology.

4.2. *Production maximization*

In the second experiment, the cost function has been tuned in order to provide the maximum production and minimize the overall energy consumption at the same time. The cost function parameters have been chosen as follows: $Q_{part} = 1$, $Q_{prod} = 2 \times 10^5$, $Q_{energy} = 1$. In contrast with the previous experiment, the production is maximized with respect to the maximum available power. The absorbed power has been dynamically constrained to a maximum value. In particular, the following bounds have been applied:

- from t=0 (min) to t=20 (min), the problem is unconstrained;
- from t=20 (min) to t=40 (min), the maximum absorbed power is $q_{max} = 4.5$ (kW);

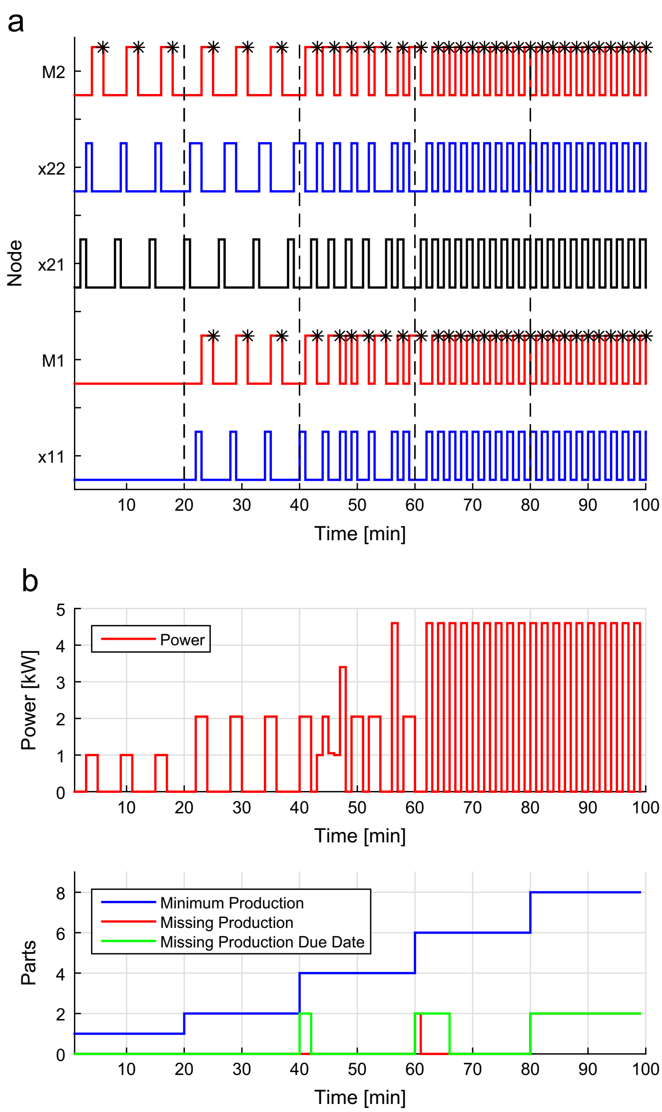

Fig. 12. Minimum production constrained problem with storage costs. (a) Machine scheduling and buffer management. The black asterisks denote the end of the machining. (b) Production and absorbed power.

- from $t=40$ (min) to $t=60$ (min), it is decreased to $q_{max}=2.2$ (kW);
- from $t=60$ (min) to $t=80$ (min), it is further decreased to $q_{max}=2.0$ (kW);
- finally, from $t=80$ (min) to $t=100$ (min), the maximum power is set to $q_{max}=1.0$ (kW) and the minimum production is increased from zero to $P_{min}=4$.

4.2.1. Simple machine model

Fig. 14 presents the results obtained with the simple machine model. The following observations are in order:

- during the first interval $t \le 20$ (min), the machines are scheduled to run at full speed. In fact, the production is maximized with higher priority than the absorbed power, which is also unbounded;
- from $t=20$ (min) to $t=40$ (min) an interesting behavior is observed: at about $t=25$ (min) the plant reaches steady state conditions where the number of produced parts is equal to the one in the previous time interval. The difference is in the management of the buffer nodes. Indeed, node $N_{2,2}$ is now always filled by a part and the two machines work alternately.

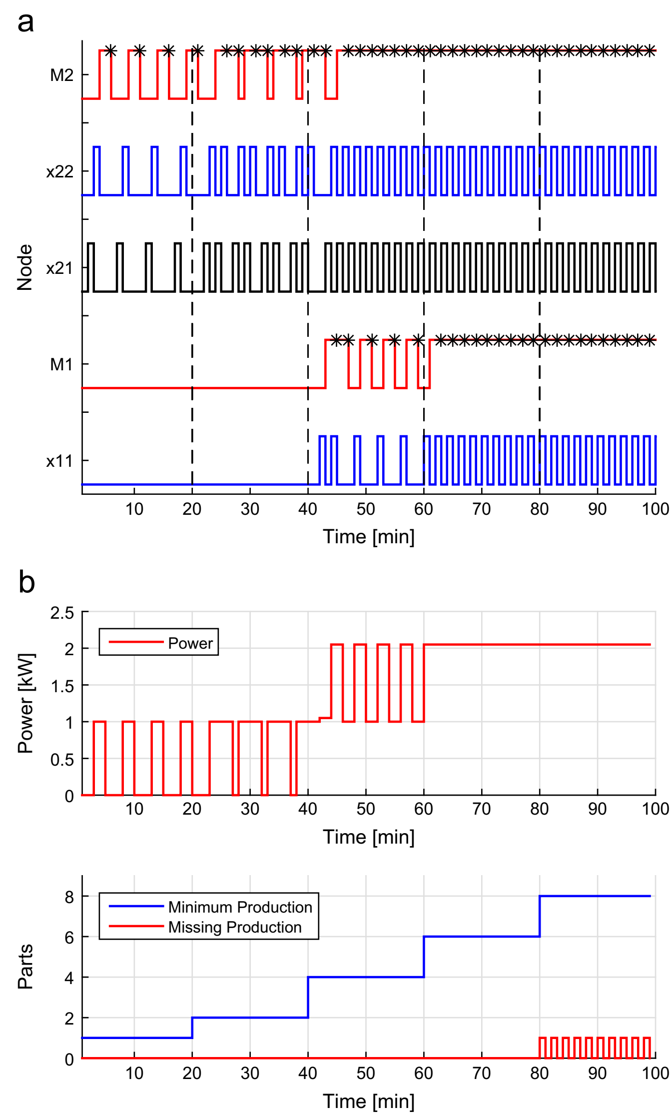

Fig. 13. Power Minimization using the Enhanced Machine Model. (a) Machine scheduling and buffer management. The black asterisks denote the end of the machining. (b) Production requirement, missing production (bottom) and absorbed power (top).

One may argue that this configuration is intuitively better than the previous one because it achieves the same production with lower energy consumption. However, this is not true due to the cost associated to the parts in the nodes, which is now higher. Different tunings of the cost function parameters may change this result;

- from $t=40$ (min) to $t=60$ (min) the upper bound of the power is further decreased and the machines must switch to slow speed production mode;
- from $t=60$ (min) to $t=80$ (min) the machines operate out of phase in order to fulfil the further reduced maximum absorbed power;
- finally, during the last interval, the maximum power is so small that the controller must operate only M_2 at slow speed. Moreover, the minimum production bound cannot be fulfilled and, since $S_p < S_q$, its violation is preferred to the power one.

4.2.2. Enhanced Machine Model

The results are illustrated in Fig. 15. The following remarks are in order:

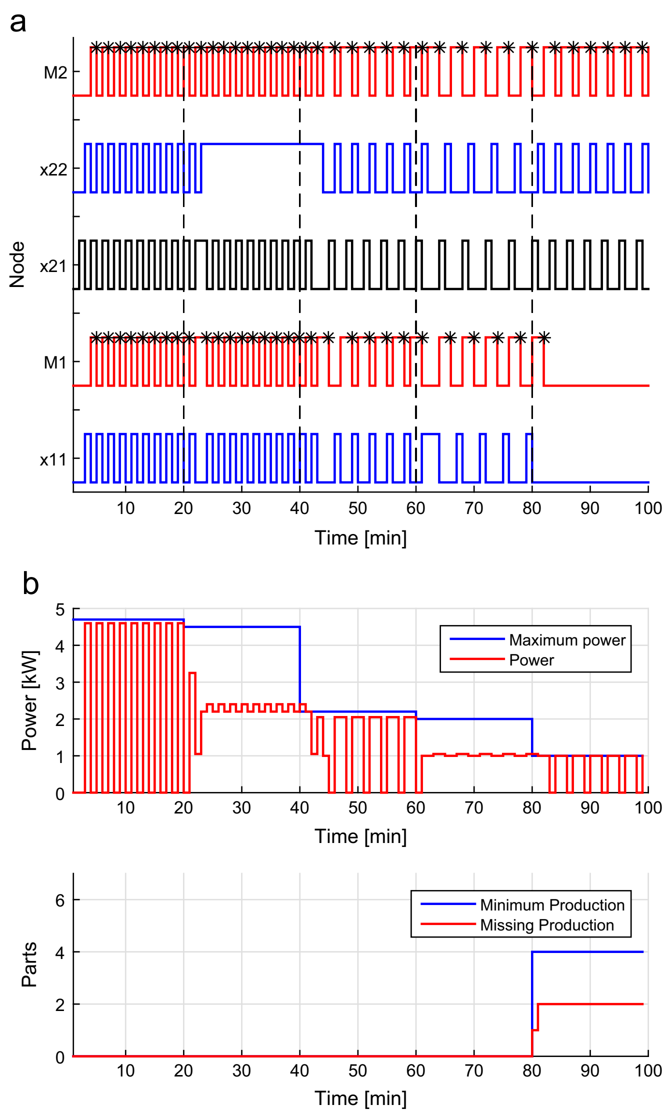

Fig. 14. Production maximization using the Simple Machine Model. (a) Machine scheduling and buffer management. The black asterisks denote the end of the machining. (b) Production and absorbed power.

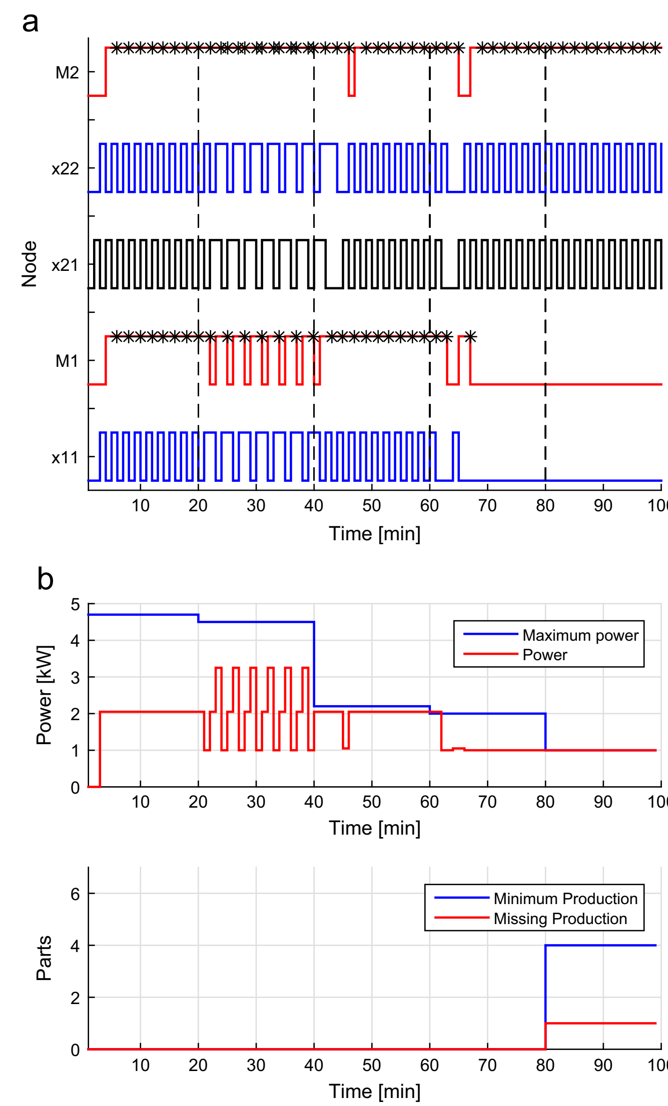

Fig. 15. Production maximization using the Enhanced Machine Model. (a) Machine scheduling and buffer management. The black asterisks denote the end of the machining. (b) Production and absorbed power.

- during the first three intervals (up to $t=60$ (min)), the machines are continuously scheduled at slow speed. In fact, as previously discussed, due to the plant topology it is not possible to continuously schedule both the machines at full speed. However, the production is equal to the previous case and the overall absorbed power is much smaller, especially in the first interval ($t<20$ (min));
- from $t=60$ (min) to $t=80$ (min), M_1, which requires an higher power, is turned off, while M_2 is continuously scheduled at slow speed;
- during the last interval, the same behavior as in the previous case is observed. However, M_2 can now continuously operate reducing the missing production.

4.3. Computational issues

The proposed algorithm requires to solve a MILP problem at any time instant; the mean values of the times required by the optimization at any time step in the simulation experiments previously described are listed in Table 2.[3] It is apparent that, in the considered example, the required computations are fully compatible with a real-time implementation. However, it is also clear that solving the optimization problem at any time step can be computationally demanding, and time required depends on the number of machines, the model adopted to describe their behavior, the number of nodes, and the length of the prediction

Table 2
Mean computational time (s) per optimization step.

Experiment	**Machine model**	
	EMM	**SMM**
Minimum production	0.400	0.303
Production maximization	0.456	0.367

[3] The simulations have been carried out on a computer Intel(R) Xeon(R) CPU E5-2620 v2 @ 2.10 GHz 2.10 GHz, 16.0 GB Installed Memory (RAM), System type 64-bit Operating System, x64-based processor, Windows 8.1 Pro., MATLAB R14b, YALMIP R20150204, CPLEX R12.4, CPLEX settings: Parallelmode=0, Threads=0.

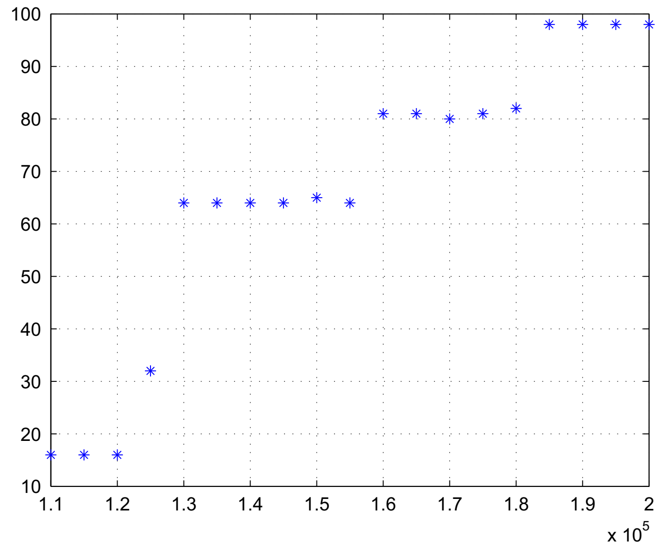

Fig. 16. Produced parts.

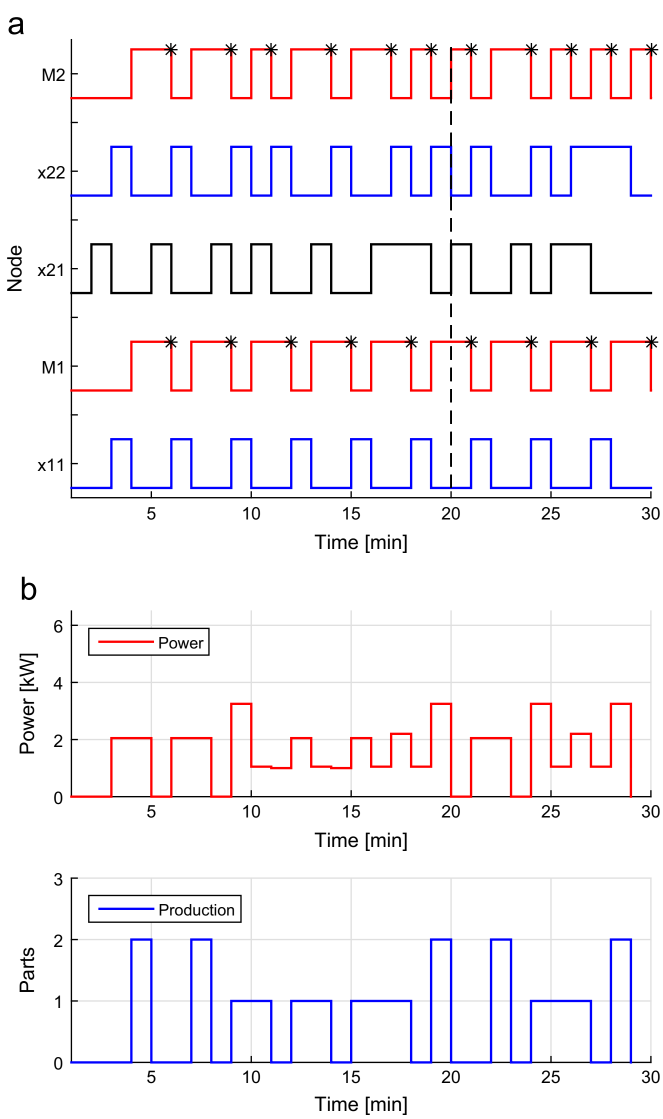

Fig. 17. Production maximization using MILP optimization. (a) Machine scheduling and buffer management. The black asterisks denote the end of the machining. (b) Production and absorbed power.

horizon. To this regard, it must be noted that also more standard MILP formulations can lead to practically unsolvable problems, see the following Section 4.5, while heuristic approaches can be very conservative. In the case of large-size problems, a possible solution consists in resorting to lagrangian relaxation techniques, which can be referred to spatial and/or temporal distributions, as already suggested in Luh and Hoitomt (1993), Raman and Grossmann (1991), and Beccuti, Geyer, and Morari (2001).

4.4. *Sensitivity to the cost function parameters*

Some experiments have been performed with the Simplified Machine Model to analyze the sensitivity of the solution with respect to the parameter Q_{prod} weighting the production in the cost function (12). To this end, it has been set $N=6$, $Q_{energy}=1$, $P_{min}=1$, $Q_{part}=0$, while Q_{prod} has been varied. The number of produced parts in the time interval $[1,100]$ is reported in Fig. 16 as a function of Q_{prod}. As expected, the curve is non decreasing and, due to the discrete nature of the production process, it is characterized by a stepwise form. Notably, when $Q_{prod} \leq 1.2 \times 10^5$ the contribution of the energy consumption dominates and the system works at the minimum production, while, if $Q_{prod} \geq 1.85 \times 10^5$, the contribution of the production sets the system to the maximum possible production value.

4.5. *Comparisons with a non RH solution*

The performances of the algorithm based on the Simplified Machine Model have been evaluated by comparing the RH implementation and the non RH solution where the goal has been to compute the control variables along a long prediction horizon of thirty time instants. The same cost function has been adopted in the two cases, with $Q_{prod}=1.2 \times 10^5$ and $Q_{energy}=1$ and $Q_{part}=10$. It has been set $P_{min}=20$ for non RH solution while the RH implementation has been computed with $N=6$, $P_{min}=4$, and including the due date constraint (18). In the non RH solution formulation the final state has been forced to be null, i.e. with empty buffers and machines, so that the computed recipe can be repeatedly applied for successive periods of the same length. The results obtained with this approach are reported in Fig. 17, while those provided by the RH implementation are shown in Fig. 18. Although the number of produced parts is the same in the two cases, the solution provided by non RH implementation is more homogeneous in terms of required power, with lower peaks. However, two comments are in order. First, the solution of the non RH problem requires 413.891 s, while the RH approach calls for a total time 9.910 s (average time of 0.330 s per step). In addition, with nonRH implementation the computational time exponentially increases, so that it is almost impossible to consider time windows larger than 35–40 sampling times. On the contrary, the computational time associated with RH is constant. Second, the RH implementation allows for much more promptness in front of possible faults of the system's components or external disturbances in view of the repeated sequence of optimizations performed at any time instant, while the non RH solution should be recomputed to cope with these perturbations. As a matter of fact, with RH a feedback control law is implemented at any sampling time, while the non RH approach leads to an open-loop control law which is prone to disturbances and/or modeling errors.

5. Conclusions

Motivated by the success of MPC in the process industry and by the increasing demand of high performing controllers in manufacturing plants, a new optimization-based control algorithm has

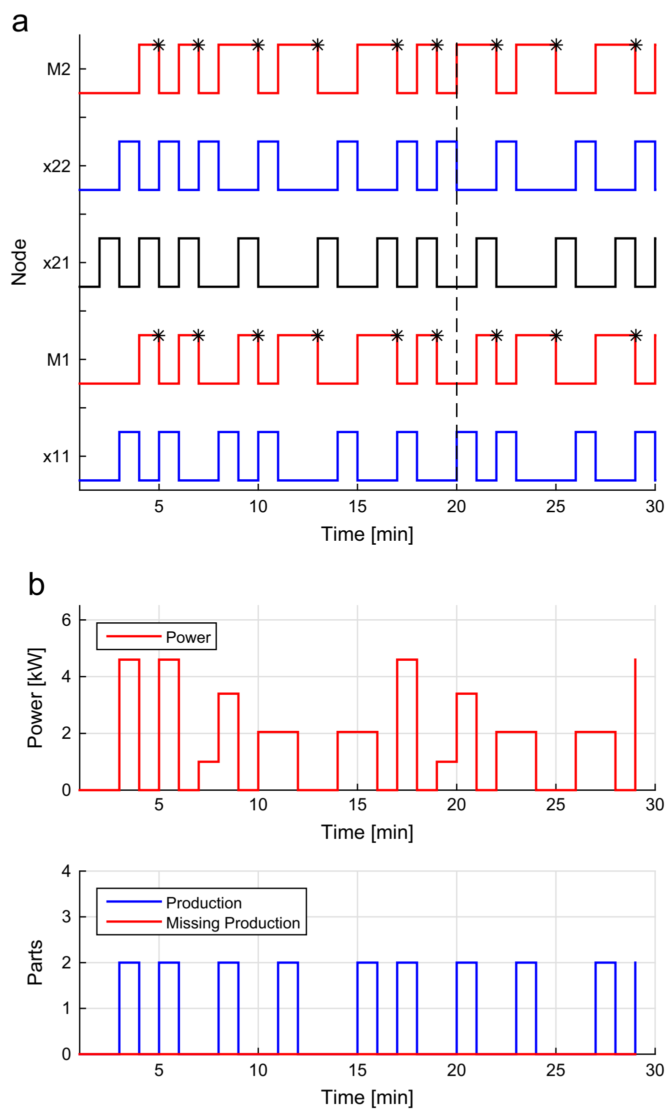

Fig. 18. Production maximization using RH optimization. (a) Machine scheduling and buffer management. The black asterisks denote the end of the machining. (b) Production and absorbed power.

been developed for the buffer management and the production scheduling of a multiple-line production plant. The simulation results reported in the paper clearly show that the algorithm can be easily adapted to obtain different behaviors by means of the tuning of simple and easy-to-understand parameters of the cost function. Moreover, the proposed method allows to cope with dynamic changes of the minimum production and maximum absorbed power and to choose the constraints to be violated in case of infeasibility. All these features are very difficult to be achieved with standard scheduling techniques based on the solution of MILP problems or on heuristics.

Many extensions of the results reported in this paper can be considered. The first, quite simple, could deal with the inclusion of constraints on the early production of parts. Other improvements could be related to the use of Lagrangian relaxation methods to simplify the optimization phase in case of large scale systems, or to the inclusion of non deterministic behaviors of the machines. For all these reasons, it is believed that model-based solutions like the one here proposed will open the way to the optimal management of high performance manufacturing plants.

Acknowledgments

This research has been supported by the Italian CNR (Consiglio Nazionale delle Ricerche) project "Genomic Model Predictive Control Tools for Evolutionary Plants" - FdF SP2-T2.1 IMET2AL.

References

Alessandri, A., Gaggero, M., & Tonelli, F. (2011). Min–max and predictive control for the management of distribution in supply chains. *IEEE Transactions on Control Systems Technology*, *19*, 1075–1089.

Angel, E., Bampis, E., & Kacem, F. (2012). Energy aware scheduling for unrelated parallel machines. In *2012 IEEE International Conference on Green Computing and Communications (GreenCom)* (pp. 533–540), November.

Beccuti, A. G., Geyer, T., & Morari, M. (2004). Temporal Lagrangian decomposition of model predictive control for hybrid systems. In *The 43rd IEEE Conference on Decision and Control*, December 14–17.

Bemporad, A., & Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, *35*(3), 407–427.

Braun, M. W., Rivera, D. E., Flores, M. E., Carlyle, W. M., & Kempf, K. G. (2003). A model predictive control framework for robust management of multi-product, multi-echelon demand networks. *Annual Reviews in Control*, *27*, 229–245.

Bruzzone, A. A. G., Anghinolfi, D., Paolucci, M., & Tonelli, F. (2012). Energy-aware scheduling for improving manufacturing process sustainability: A mathematical model for flexible flow shopson a generalized approach to manufacturing energy efficiency. *CIRP Annals—Manufacturing Technology*, 459–462.

Camacho, E. F., & Bordons Alba, C. (2004). *Model Predictive Control* (2nd edition). London: Springer.

Cataldo, A., & Scattolini, R. (2014). Logic control design and discrete event simulation model implementation for a de-manufacturing plant. *Automazione-plus*.

Cataldo, A., & Scattolini, R. (2014). Modeling and model predictive control of a de-manufacturing plant. In *2014 IEEE International Conference on Control Applications (CCA) – Part of 2014 IEEE Multi-conference on Systems and Control* (pp. 1855–1860), Antibes Congress Center, Nice/Antibes, France, October 8–10.

Colledani, M., Copani, G., & Tolio, T. (2014). De-manufacturing systems. In *Variety Management in Manufacturing. Proceedings of the 47th CIRP Conference on Manufacturing Systems* (pp. 14–19), Windsor, Ontario, Canada, April 28–30.

Copani, G., Brusaferri, A., Colledani, M., Pedrocchi, N., Sacco, M., & Tolio, T. (2012). Integrated de-manufacturing systems as new approach to end-of-Life management of mechatronic devices. In *10th Global Conference on Sustainable Manufacturing Towards Implementing Sustainable Manufacturing*, Istanbul, Turkey.

Dai, M., Tang, D., Giret, A., Salido, M. A., & Li, W.D. (2013). Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. *Robotics and Computer-Integrated Manufacturing*, 418–429.

European Commission (2009). ICT and energy efficiency—the case for manufacturing. *Recommendations of the Consultation Group, European Commission*, no. ISBN: 978-92-79-11306-2.

Fang, K., Uhan, N., Zhao, F., & Sutherland, J. W. (2011). A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. *Journal of Manufacturing Systems*, 234–240.

Ferrio, J., & Wassick, J. (2008). Chemical supply chain network optimization. *Computers and Chemical Engineering*, *32*, 2481–2504.

Fysikopoulos, A., Pastras, G., Alexopoulos, T., & Chryssolouris, G. (2014). On a generalized approach to manufacturing energy efficiency. *The International Journal of Advanced Manufacturing Technology*, 1–16.

Haït, A., & Artigues, C. (2009). Scheduling parallel production lines with energy costs. In *The 13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM09)*, June.

Heemels, W. P. M. H., De Schutter, B., & Bemporad, A. (2001). Brief equivalence of hybrid dynamical models. *Automatica*, *37*(July (7)), 1085–1091.

Herceg, M., Kvasnica, M., Jones, C. N., & Morari, M. (2013). Multi-Parametric Toolbox 3.0. In *Procedings of the European Control Conference* (pp. 502–510). Zürich, Switzerland, July 17–19.

ICF International (2007). Energy trends in selected manufacturing sectors: Opportunities and challenges for environmentally preferable energy outcomes. *Final Report*.

Jansen, K., & Porkolab, L. (2001). Improved approximation schemes for scheduling unrelated parallel machines. *Mathematics of Operations Research*, *26*(2), 324–338.

Kim, D. W., Na, D. G., & Chen, F. (2003). Unrelated parallel machine scheduling with setup times and a total weighted tardiness objective. *Robotics and Computer-Integrated Manufacturing*, *19*(1–2), 173–181.

Lucas, C., Mitra, G., & Moody, S. (1992). Tools for reformulating logical forms into zero-one mixed integer programs (MIPS). *Maths Technical Papers (Brunel University) (TR/03/92)* (pp. 1–27), April.

Luh, P. B., & Hoitomt, D. J. (1993). Scheduling of manufacturing systems using the Lagrangian relaxation technique. *IEEE Transactions on Automatic Control*, 1066–1079.

McKinnon, K. I. M., & Williams, H. P. (1989). Constructing integer programming models by the predicate calculus. *Annals of Operations Research*(21), 227–246.

Organization for Economic Cooperation and Developoment (2004). OECD Key Environmental Indicators.
Pinedo, M. L. (2008). *Scheduling*. New York, NY: Springer.
Rabadi, G., Moraga, R. J., & Al-Salem, A. (2006). Heuristics for the unrelated parallel machine scheduling problem with setup times. *Journal of Intelligent Manufacturing*, *17*(1), 85–97.
Raman, R., & Grossmann, I. E. (1991). Relation between MILP modeling and logical inference for chemical process synthesis. *Computers and Chemical Engineering*, *15*(2), 73–84.
Seliger, G. (Ed.). (2007). *Sustainability in Manufacturing*. Berlin: Springer.
Senthilkumar, P., & Narayanan, S. (2010). Literature review of single machine scheduling problem with uniform parallel machines. *Intelligent Information Management*, *2*(8), 457–474.
Torrisi, F. D., & Bemporad, A. (2004). HYSDEL—A tool for generating computational hybrid models for analysis and synthesis problems. *IEEE Transactions on Control Systems Technology*, *12*(March), 235–249.
Vargas-Villamir, F. D., & Rivera, D. E. (2000). Multilayer optimization and scheduling using model predictive control: *application to reentrant semiconductor manufacturing lines*. *Computers and Chemical Engineering*, *24*(8), 2009–2021.
Vargas-Villamir, F. D., & Rivera, D. E. (2001). A model predictive control approach for real-time optimization of reentrant manufacturing lines. *Computers in Industry*, *45*(1), 45–57.
Wang, W., & Rivera, D. E. (2008). Model predictive control for tactical decision-making in semiconductor manufacturing supply chain management. *IEEE Transactions on Control Systems Technology*, *16*, 841–855.
Weng, M. X., Lu, J., & Ren, H. (2001). Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective. *Interna-tional Journal of Production Economics*, *70*(3), 215–226.
Williams, H. P. (2013). *Model Building in Mathematical Programming* (5th edition). Chichester, West Sussex, England: Wiley.
Ydstie, B. E. (2004). Distributed decision making in complex organizations: *the adaptive enterprise*. *Computers and Chemical Engineering*, *29*, 11–27.