Models and Solving Procedures for Continuous-Time Production Planning

Stéphane Dauzère-Pérès[†] Stanley B. Gershwin[‡]
Marc Sevaux[†]

[†]Département Automatique et Productique
Ecole des Mines de Nantes
4, rue A. Kastler – BP 20722
44307 Nantes cedex 3
FRANCE

[‡]Laboratory for Manufacturing and Productivity
Department of Mechanical Engineering
Massachusetts Institute of Technology
77, Massachusetts Avenue
Cambridge, MA 02139
USA

July 20, 1998

Abstract

The goal of this paper is to solve a continuous time production planning problem, where production rates are supposed to be piecewise constant. The problem is then to determine not only the rates, but also the times, called switching times, at which these rates may change. An iterative procedure is proposed where, at the first step, optimal rates are determined given the switching times and, at the second step, new switching times are added given the rates.

Various models are proposed and discussed for the first step, and a linear programming formulation is finally retained. A special effort has been made to reduce the size of the linear programming problem solved at each iteration of the procedure. Several rules to remove or add switching times are also analyzed. Computational experiments are presented comparing two versions of the procedure and showing its efficiency.

1 Introduction

In classical planning problems, based on the demand forecast on a given horizon, the objective is to determine production quantities in order to minimize surplus, *i.e.*, the difference between cumulative production and cumulative demand, for each product. Surplus can be positive (*inventory*) or negative (*backlog*). Usually, the horizon is discretized into periods of equal lengths, and surpluses are only considered at the end of each period (see [2]). The problem is by definition a discrete time problem.

In industries where items are produced in large quantities (such as automobiles, paper, food, and semiconductors), raw material is continuously arriving and final products are continuously leaving the shop-floor. Moreover, demands are often continuously taken from the inventories, and not only at ends of equally spaced periods as it is assumed in discrete-time models. This phenomenon is emphasized at the tactical level (on a 6 to 18 month horizon), where arrivals of lots of products every few hours in the inventory can also be seen as continuous. Thus, the problem can be stated in terms of production and demand rates. The objective is to minimize an integral of surplus cost over the time horizon. In this paper, we want to solve this **continuous-time production planning** problem. Demand rates are assumed to be known and piecewise constant (although our results could be extended to the case where the demand is discrete). Accordingly, in our models, the time horizon is broke up into periods which may have different lengths. Time periods are defined as periods during which the demand rate is constant. The instantaneous capacity of the system is sometimes not sufficient for the instantaneous demand rates, in particular when demand is very large for a time-consuming product. Hence, in other periods, it is necessary to produce more than the current demands. The problem is to determine how much more, and when and how long to produce it.

To reduce the cost, it is actually interesting to allow production rates to change at other time instants than ends of periods. All these instants, including the ends of periods, are called *switching times*. Hence, one has to determine both the production rates and the switching times. Because it is very difficult to determine them simultaneously, a two-step iterative procedure is proposed. In the first step, given the switching times, optimal production rates are determined. The size of the problem mainly depends on the number of switching times. These times are important when they represent natural changes of the production rates. Consequently, rather than having a lot of switching times (*i.e*, a lot of variables), it is more important to accurately place these times. This allows the size of the problem to be greatly reduced. Hence, in the second step of the procedure, given the rates, redundant switching times are removed and new switching times are added. This is done such that the cost will decrease at the next iteration. The procedure iterates between the two steps until there is no significant improvement during two consecutive iterations.

Our results can also be useful in a hierarchical framework such as the one developed in [6] where the decision variables are the production rates. The demand is assumed constant, and the objective is to determine a policy to minimize inventory and backlog over an infinite horizon in the stochastic case, *i.e.*, when uncontrollable and unpredictable events such as failures and repairs of machines are taken into account. The solution is a piecewise constant production rate. Here, we consider the deterministic case (reliable machines) without setup times, and with time-varying demand rates. Our goal is to add, most likely at the highest level, a scheduler in the hierarchy proposed by [7]. Production rates computed by this module become targets for schedulers in the lower levels.

Our problem can be seen as a special case of the general framework proposed by Hachman and Leachman [8], or of the general model of Sharifnia [9]. However, in these two papers, the authors are more interested in modeling of production planning and scheduling than in the actual solving of the model. In particular, by using time discretization, Sharifnia states that the main advantage of his formulation is the "efficiency of LP codes for solving large problems". As we shall see in the numerical experiments of Section 7.3, the CPU time can be prohibitive if the number of products or periods is too large.

The problem is presented and the two-step iterative procedure is sketched in Section 2. In Section 3, two models are introduced for the first step of the procedure. After deriving the expression of the exact cost, a nonlinear programming model (NLP) is presented in Section 3.2, where the associated cost J_{NLP} approximates the exact cost. Then, a linear programming model (LP) is derived by only keeping the linear part of the cost. Properties of the LP model are discussed in Section 3.3. In Section 4, several propositions are presented, from which we derive five rules to add and remove switching times. These rules are used in the second step of the iterative procedure, to modify the set of switching times (*given* the production rates) such that the exact cost is ensured not to increase at the next iteration. Two versions of the iterative procedure are detailed in Section 5. Section 6 shows how the number of variables can be reduced at every step of the procedure. Some numerical experiments are presented and discussed in Section 7.

2 The production planning problem

2.1 The continuous time model

Let n be the number of part types (or products), m the number of resources (or machines), and C the planning horizon where demand for each product is known. We introduce the following notation :

$M = [1, .., m]$: set of resources
$d_i(t)$: demand rate of part type i at time t
$u_i(t)$: production rate of part type i at time t
τ_{ik}: average processing time per unit of product i on resource k
$x_i(t)$: surplus level of product i at time t (x_{i0} initial surplus, and $x_i(t_1) = x_{i0} + \int_0^{t_1}(u_i(t) - d_i(t))dt$).

Information about failures, repairs and set-up times on the resources may be considered in the value of the processing times. However, these phenomena are not explicitly taken into account in this model.

Capacity constraints on rates $u_i(t)$ are:

$$\sum_i \tau_{ik} u_i(t) \leq 1 \quad \forall k \in M;\ 0 \leq t \leq C$$

which states that each resource cannot work more than 100% of its time.

Consider the problem in which we want to plan the production of the n products over the planning horizon C. The usual objective is to minimize the following cost function, depending on the production rates u and the surpluses x:

$$\sum_{i=1}^{n} \int_0^C g_i(x_i(t))dt = \sum_{i=1}^{n} \int_0^C [h_i^+ x_i^+(t) + h_i^- x_i^-(t)]dt$$

where $x_i(t) = x_i^+(t) - x_i^-(t)$ ($x_i^+(t)$, $x_i^-(t) \geq 0$) is the surplus at time t. $x_i^+(t)$ is the inventory and $x_i^-(t)$ the backlog. h_i^+ and h_i^- are holding cost and backlog cost per product and **time** unit. A production cost q_i can be considered in the objective function, by adding the term $q_i u_i(t)$ to $g_i(x_i(t))$, and all the results in the remainder of the paper can be rather easily extended.

When demand rates are known in advance over a horizon $[0, C]$, the following problem has to be solved:

$$P^{ct} \begin{cases} \min J = \sum_{i=1}^{n} \int_0^C g_i(x_i(t))dt & & (1) \\ \dot{x}_i(t) = u_i(t) - d_i(t) & i = 1,..,n & (2) \\ x_i(0) = x_{i0} & i = 1,..,n & (3) \\ \sum_{i=1}^{n} \tau_{ik} u_i(t) \leq 1 & \forall k \in M & (4) \end{cases}$$

where x_{i0} is the initial surplus.

P^{ct} is a deterministic dynamic programming problem with continuous time and state variables. Bellman's equation for this problem is a nonlinear partial differential equation that is usually impossible to solve in practice (in particular when d_i is a function of time) [7].

In the remainder of this work, we shall assume that the planning horizon $[0, C]$ is divided into T periods, during which the demand is constant for every product. Hence, periods may have different lengths. The length of a period l ($l \in [1,..,T]$) is given and denoted by c_l, and its ending time by t_l (*i.e.*, $t_l - t_{l-1} = c_l$, $t_0 = 0$, and $t_T = C$). Consequently, $d_i(t) = d_{il}$ $\forall i, \forall t \in [t_{l-1}, t_l]$.

In the following section, the problem is formulated assuming that production rates are constant on each period ($u_i(t) = u_{il}$, $\forall t \in [t_{l-1}, t_l]$), *i.e.*, $u_i(t)$ is piecewise constant over the horizon.

2.2 The problem with constant production rates

Because production rates are constant on each period of the horizon, $x_i(t)$ is a piecewise linear function, and (2) is:

$$x_i(t) = \int_0^t (u_i(\theta) - d_i(\theta))d\theta = x_{i0} + \int_{t_0}^{t_1} (u_{i1} - d_{i1})d\theta + ... + \int_{t_{l-2}}^{t_{l-1}} (u_{il-1} - d_{il-1})d\theta + \int_{t_{l-1}}^{t} (u_{il} - d_{il})d\theta$$

where l is such that $t_{l-1} \leq t \leq t_l$.

Then, since $t_l - t_{l-1} = c_l$,

$$x_i(t) = x_{i0} + \sum_{j=1}^{l-1} (u_{ij} - d_{ij})c_j + (u_{il} - d_{il})(t - t_{l-1}) \quad i = 1,..,n, \tag{5}$$

Let x_{il} denote $x_i(t_l)$. Then

$$x_{il} = x_{i0} + \sum_{j=1}^{l} (u_{ij} - d_{ij})c_j$$

$$x_{il} = x_{il-1} + (u_{il} - d_{il})c_l \quad i = 1,..,n; \; l = 1,..,T$$

and

$$u_{il} - d_{il} = \frac{x_{il} - x_{il-1}}{c_l} \quad i = 1,..,n; \; l = 1,..,T, \text{ and } c_l > 0 \tag{6}$$

(1) can be written:

$$\min J = \sum_{i=1}^{n} \sum_{l=1}^{T} \int_{t_{l-1}}^{t_l} g_i(x_i(t))dt$$

and, by using (5),

$$\min J = \sum_{i=1}^{n} \sum_{l=1}^{T} \int_{t_{l-1}}^{t_l} g_i(x_{il-1} + (u_{il} - d_{il})(t - t_{l-1}))dt$$

P^{ct} is now:

$$\begin{cases} \min J = \sum_{i=1}^{n} \sum_{l=1}^{T} \int_{t_{l-1}}^{t_l} g_i(x^+_{il-1} - x^-_{il-1} + (u_{il} - d_{il})(t - t_{l-1}))dt & & (7) \\ x^+_{il} - x^-_{il} \quad = x^+_{il-1} - x^-_{il-1} + (u_{il} - d_{il})c_l & i = 1,..,n;\ l = 1,..,T & (8) \\ \sum_{i=1}^{n} \tau_{ik} u_{il} \quad \leq 1 & \forall k \in M;\ l = 1,..,T & (9) \\ u_{il}, x^+_{il}, x^-_{il} \quad \geq 0 & i = 1,..,n;\ l = 1,..,T & (10) \end{cases}$$

As shown in Figure 1 (for Product i), we want to minimize the weighted area between the cumulative demand and production lines. In our continuous-time production planning problem, compared to a classical discrete time approach where backlog and inventory costs are per product unit, costs are per product *and time* unit. Thus, different period lengths c_l are naturally taken into account in the objective J.

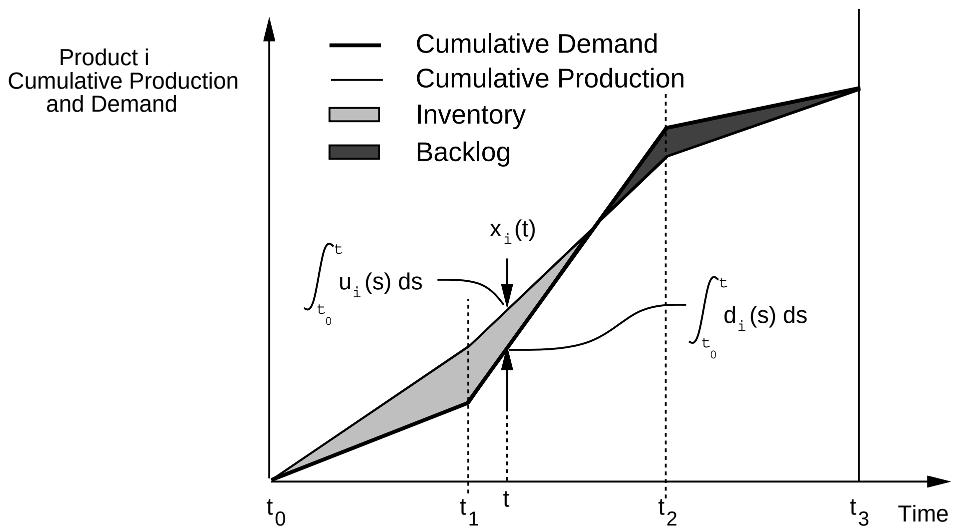

Figure 1: Inventory and backlog

The precise expression of Criterion (7) is computed in the next section, and it is nonlinear in x. There is no reason why production should only change when demand is changing. Moreover, by allowing production rates to change during periods where demands stay constant, one can expect to greatly reduce the cost. A rate will be allowed to *switch* from one value to another at given times called *switching times*, and has to remain constant between them. The ending times of periods (t_l) will also be considered as switching times. In Figure 2, a switching time has been

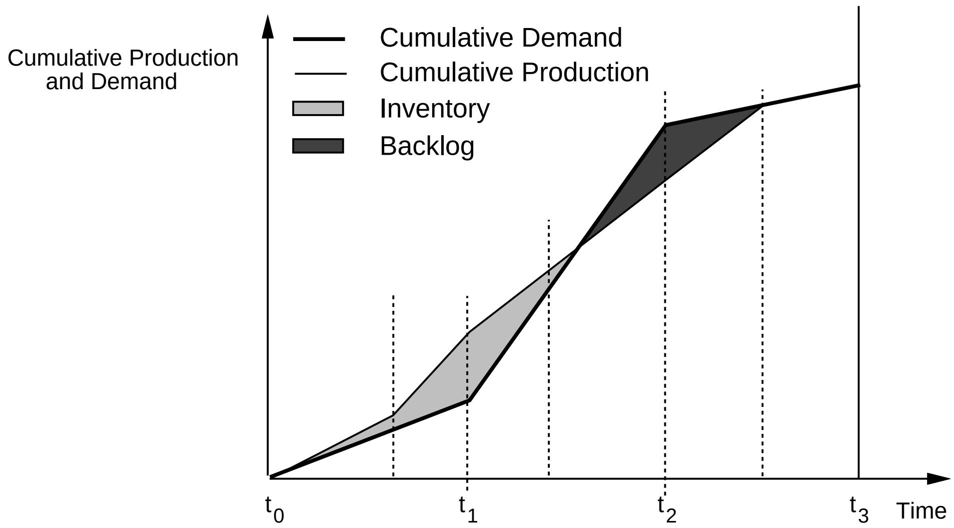

Figure 2: Inventory and backlog with switching times

fixed arbitrarily in each period. The production rate changes in the first period ($[t_0, t_1]$) in order to follow the demand more closely, and in the third period ($[t_2, t_3]$) to reach the demand earlier and avoid some backlog.

$S_l - 1$ is the number of switching times within each period l. Let t_{ls} ($s \in [1, S_l]$) denote the s^{th} switching time in period l, and c_{ls} be $t_{ls} - t_{ls-1}$, *i.e.*, $\sum_{s=1}^{S_l} c_{ls} = c_l$. We define $t_{l0} = t_{l-1}$ and $t_{lS_l} = t_l$. Surpluses and production rates have to be computed at each switching time. And if lengths of periods are data of the problem, switching times are actually variables (S_l, t_{ls} and c_{ls}).

Because determining simultaneously production rates and switching times is a very difficult nonlinear optimization problem, we propose an iterative procedure (see Figure 3). At the first level, with fixed switching times, production rates are optimized, *i.e.*, optimal values for u_{ils} and x_{ils} are determined. Then, at the second level, given the production rates, various rules are used to change the set of switching times, and obtain new values for S_l and c_{ls}. The problem to solve at the first step is discussed in Section 3. The propositions used to derive the rules of the second step are presented in Section 4.

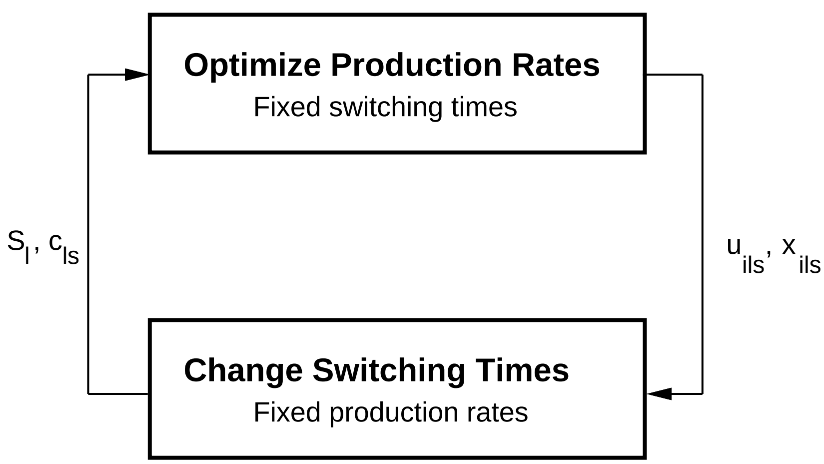

Figure 3: The iterative procedure

3 Optimizing production rates

The *exact* cost J, explicitly determined in Section 3.1, is not a single analytical expression on the whole surplus space. In Section 3.2, a solvable Nonlinear Programming (NLP) model is proposed by approximating J by a simpler nonlinear cost J_{NLP}. Then, by suppressing the nonlinear part in the exact cost, a Linear Programming (LP) model (cost J_{LP}) is derived in Section 3.3, and advantages and drawbacks of this model are discussed. Finally, the LP model is compared with a classical discrete time formulation in Section 3.4.

3.1 The exact cost

In order to explicitly write the criterion, we examine, for a given product i, a given period l and a given switching time s, the following cases:

1. $x_{ils-1}, x_{ils} \geq 0$, *i.e.* $x^+_{ils-1}, x^+_{ils} \geq 0$, and $x^-_{ils-1} = x^-_{ils} = 0$.
2. $x_{ils-1}, x_{ils} \leq 0$, *i.e.* $x^-_{ils-1}, x^-_{ils} \geq 0$, and $x^+_{ils-1} = x^+_{ils} = 0$.
3. $x_{ils-1} > 0$, $x_{ils} < 0$, *i.e.* $x^+_{ils-1}, x^-_{ils} > 0$, and $x^-_{ils-1} = x^+_{ils} = 0$.
4. $x_{ils-1} < 0$, $x_{ils} > 0$, *i.e.* $x^-_{ils-1}, x^+_{ils} > 0$, and $x^+_{ils-1} = x^-_{ils} = 0$.

This is because, as we shall see, the expression for

$$J_{ils} = \int_{t_{ls-1}}^{t_{ls}} g_i(x_{ils-1} + (u_{ils} - d_{il})(t - t_{ls-1}))dt \qquad (11)$$

is different for each case.

An example of the first case can be seen in $[t_0, t_1]$ of Figure 1, of the second case in $[t_2, t_3]$, and of the third case in $[t_1, t_2]$.

Case 1

Since $x_i(t)$ is positive on the interval $[t_{ls-1}, t_{ls}]$ (*i.e.*, $x_i(t) = x^+_i(t)$), $g_i(x_i(t)) = h^+_i x^+_i(t)$ $\forall t \in [t_{ls-1}, t_{ls}]$, and J_{ils} becomes

$$J^1_{ils} = \int_{t_{ls-1}}^{t_{ls}} h^+_i(x^+_{ils-1} + (u_{ils} - d_{il})(t - t_{ls-1}))dt$$

$$= h^+_i \left(x^+_{ils-1}(t_{ls} - t_{ls-1}) + \frac{u_{ils} - d_{il}}{2}(t_{ls} - t_{ls-1})^2 \right) = h^+_i c_{ls} \left(x^+_{ils-1} + \frac{c_{ls}(u_{ils} - d_{il})}{2} \right)$$

and, with (6),

$$J^1_{ils} = \frac{h^+_i c_{ls}(x^+_{ils-1} + x^+_{ils})}{2}$$

This do not cause any difficulties because J^1_{ils} is linear in x^+_{ils-1} and x^+_{ils}.

Case 2

Here, $x_i(t)$ is negative on the interval $[t_{ls-1}, t_{ls}]$ (*i.e.*, $x_i(t) = -x_i^-(t)$), then $g_i(x_i(t)) = h_i^- x_i^-(t)$ $\forall t \in [t_{ls-1}, t_{ls}]$, and

$$J_{ils}^2 = \int_{t_{ls-1}}^{t_{ls}} h_i^-(x_{ils-1}^- - (u_{ils} - d_{il})(t - t_{ls-1}))dt$$

and, similarly to Case 1,

$$J_{ils}^2 = \frac{h_i^- c_{ls}(x_{ils-1}^- + x_{ils}^-)}{2}$$

This case also causes no difficulties.

Case 3

In this case, $x_i(t)$ is positive between t_{ls-1} and some time t_*, and negative between t_* and t_{ls}. Here, t_* is the time where the cumulative production line crosses the cumulative demand line (see period 2 in Figure 1), *i.e.*, $x_i(t_*) = 0$. Plugging this in (5), we have

$$x_i(t_*) = 0 = x_{ils-1}^+ + (u_{ils} - d_{il})(t_* - t_{ls-1})$$

or,

$$t_* = t_{ls-1} - \frac{x_{ils-1}^+}{u_{ils} - d_{il}} \tag{12}$$

Also, for $t \in [t_*, t_{ls}]$,

$$x_i(t) = -x_i^-(t) = x_{ils-1}^+ + (u_{ils} - d_{il})(t - t_* + t_* - t_{ls-1}) = (u_{ils} - d_{il})(t - t_*)$$

In particular,

$$x_i(t_{ls}) = -x_{ils}^- = (u_{ils} - d_{il})(t_{ls} - t_*)$$

so

$$t_* = t_{ls} + \frac{x_{ils}^-}{u_{ils} - d_{il}} \tag{13}$$

J_{ils}^3 is equal to

$$\int_{t_{ls-1}}^{t_*} g_i(x_{ils-1}^+ + (u_{ils} - d_{il})(t - t_{ls-1}))dt + \int_{t_*}^{t_{ls}} g_i((u_{ils} - d_{il})(t - t_*))dt$$

$$= h_i^+[x_{ils-1}^+(t_* - t_{ls-1}) + (u_{ils} - d_{il})(\frac{t_*^2}{2} - t_{ls-1}t_* - \frac{t_{ls-1}^2}{2} + t_{ls-1}^2)]$$

$$+h_i^-[-(u_{ils} - d_{il})(\frac{t_{ls}^2}{2} - t_*t_{ls} - \frac{t_*^2}{2} + t_*^2)]$$

and, with (12) and (13),

$$J_{ils}^3 = -\frac{h_i^+(x_{ils-1}^+)^2}{2(u_{ils}-d_{il})} - \frac{h_i^-(x_{ils}^-)^2}{2(u_{ils}-d_{il})}$$

and, with (6),

$$J_{ils}^3 = \frac{c_{ls}}{2}\frac{h_i^+(x_{ils-1}^+)^2 + h_i^-(x_{ils}^-)^2}{x_{ils-1}^+ + x_{ils}^-}$$

This do cause difficulties because J_{ils}^3 is not a linear function.

Case 4

Following the same steps as in Case 3,

$$J_{ils}^4 = \frac{c_{ls}}{2}\frac{h_i^-(x_{ils-1}^-)^2 + h_i^+(x_{ils}^+)^2}{x_{ils-1}^- + x_{ils}^+}$$

Similarly, this case leads to difficulties.

In summary,

$$J = \sum_{i=1}^{n}\sum_{l=1}^{T}\sum_{s=1}^{S_l} J_{ils} \tag{14}$$

where

$$\left\| \begin{array}{ll} J_{ils} = J_{ils}^1 = \frac{h_i^+ c_{ls}(x_{ils-1}^+ + x_{ils}^+)}{2} & \text{if } x_{ils-1}^+, x_{ils}^+ \geq 0, \text{ and } x_{ils-1}^- = x_{ils}^- = 0 \\ J_{ils} = J_{ils}^2 = \frac{h_i^- c_{ls}(x_{ils-1}^- + x_{ils}^-)}{2} & \text{if } x_{ils-1}^-, x_{ils}^- \geq 0, \text{ and } x_{ils-1}^+ = x_{ils}^+ = 0 \\ J_{ils} = J_{ils}^3 = \frac{c_{ls}}{2}\frac{h_i^+(x_{ils-1}^+)^2 + h_i^-(x_{ils}^-)^2}{x_{ils-1}^+ + x_{ils}^-} & \text{if } x_{ils-1}^+, x_{ils}^- > 0, \text{ and } x_{ils-1}^- = x_{ils}^+ = 0 \\ J_{ils} = J_{ils}^4 = \frac{c_{ls}}{2}\frac{h_i^-(x_{ils-1}^-)^2 + h_i^+(x_{ils}^+)^2}{x_{ils-1}^- + x_{ils}^+} & \text{if } x_{ils-1}^-, x_{ils}^+ > 0, \text{ and } x_{ils-1}^+ = x_{ils}^- = 0 \end{array} \right.$$

Let us derive some results that will be useful in the sequel.

Remark 1

(a) $J_{ils}^1 = J_{ils}^3 + J_{ils}^4$, *if* $x_{ils-1}^+,\ x_{ils}^+ > 0$.

(b) $J_{ils}^2 = J_{ils}^3 + J_{ils}^4$, *if* $x_{ils-1}^-,\ x_{ils}^- > 0$.

(c) $\lim_{x_{ils-1}^+ \to 0, x_{ils}^- \to 0} J_{ils}^3 = \lim_{x_{ils-1}^- \to 0, x_{ils}^+ \to 0} J_{ils}^4 = 0$,

Proof:

In (a), since $x_{ils-1}^+,\ x_{ils}^+ > 0$ ($\Rightarrow x_{ils-1}^- = x_{ils}^- = 0$), expressions J_{ils}^3 and J_{ils}^4 are defined. $J_{ils}^3 = (h_i^+ x_{ils-1}^+)c_{ls}/2$, and $J_{ils}^4 = (h_i^+ x_{ils}^+)c_{ls}/2$, hence $J_{ils}^1 = J_{ils}^3 + J_{ils}^4$. (b) can be shown in the same way. Finally, (c) comes from the fact that the numerator in J_{ils}^3 and J_{ils}^4 tends to 0 faster than the denominator. □

3.2 A NonLinear Programming (NLP) model

Based on Remark 1 and as demonstrated in Proposition 1, J_{ils} can be approximated by

$$J'_{ils} = J^{3'}_{ils} + J^{4'}_{ils} = \frac{c_{ls}}{2}\left[\frac{h^+_i(x^+_{ils-1})^2 + h^-_i(x^-_{ils})^2}{\max(\epsilon, x^+_{ils-1} + x^-_{ils})}\right] + \frac{c_{ls}}{2}\left[\frac{h^-_i(x^-_{ils-1})^2 + h^+_i(x^+_{ils})^2}{\max(\epsilon, x^-_{ils-1} + x^+_{ils})}\right] \quad (15)$$

where ϵ is strictly positive and small enough.

Proposition 1

$$J_{ils} \geq J'_{ils}, \quad i = 1,..,n, \ l = 1,..,T, \ s = 1,..,S_l$$

Moreover, the difference between J_{ils} and J'_{ils} is bounded by $c_{ls}(h^+_i + h^-_i)\epsilon$.

Proof:

When x^+_{ils-1}, $x^+_{ils} \geq \epsilon$ or x^-_{ils-1}, $x^-_{ils} \geq \epsilon$, $J'_{ils} = J^3_{ils} + J^4_{ils} = J_{ils}$ by Remark 1. $J'_{ils} = J^3_{ils} = J_{ils}$ when $x^+_{ils-1} + x^-_{ils} \geq \epsilon$ and $x^-_{ils-1} + x^+_{ils} = 0$. $J'_{ils} = J^4_{ils} = J_{ils}$ when $x^-_{ils-1} + x^+_{ils} \geq \epsilon$ and $x^+_{ils-1} + x^-_{ils} = 0$. Finally, $J'_{ils} = J_{ils} = 0$ when $x^+_{ils-1} = x^-_{ils-1} = x^+_{ils} = x^-_{ils-1} = 0$.

Actually $J'_{ils} \neq J_{ils}$ only when $0 < x^+_{ils-1} + x^-_{ils} < \epsilon$, $0 < x^-_{ils-1} + x^+_{ils} < \epsilon$, or both conditions hold. Let us consider these three cases.

i) $0 < x^+_{ils-1} + x^-_{ils} < \epsilon$, and $x^-_{ils-1} + x^+_{ils} = 0$ ($J_{ils} = J^3_{ils}$, $J^{4'}_{ils} = 0$) or $x^-_{ils-1} + x^+_{ils} \geq \epsilon$ ($J_{ils} = J^3_{ils} + J^4_{ils}$, $J^{4'}_{ils} = J^4_{ils}$). Then, because $J^{3'}_{ils} > 0$,

$$J_{ils} - J'_{ils} = J^3_{ils} - J^{3'}_{ils} < J^3_{ils}$$

$$J_{ils} - J'_{ils} < \frac{c_{ls}}{2}\frac{h^+_i(x^+_{ils-1})^2 + h^-_i(x^-_{ils})^2}{x^+_{ils-1} + x^-_{ils}} = \frac{c_{ls}}{2}\frac{h^+_i(x^+_{ils-1} + x^-_{ils})^2 + h^-_i(x^+_{ils-1} + x^-_{ils})^2}{x^+_{ils-1} + x^-_{ils}}$$

$$J_{ils} - J'_{ils} < \frac{c_{ls}}{2}(h^+_i(x^+_{ils-1} + x^-_{ils}) + h^-_i(x^+_{ils-1} + x^-_{ils})) < \frac{c_{ls}}{2}(h^+_i + h^-_i)\epsilon$$

since $x^+_{ils-1} + x^-_{ils} < \epsilon$.

ii) $0 < x^-_{ils-1} + x^+_{ils} < \epsilon$, and $x^+_{ils-1} + x^-_{ils} = 0$ or $x^+_{ils-1} + x^-_{ils} \geq \epsilon$. Then, as in *i)*,

$$g_{il} - g'_{il} < \frac{c_l}{2}(h^+_i + h^-_i)\epsilon$$

iii) $0 < x^+_{ils-1} + x^-_{ils} < \epsilon$, and $0 < x^-_{ils-1} + x^+_{ils} < \epsilon$. Because $J'_{ils} > 0$,

$$J_{ils} - J'_{ils} < J_{ils} = \frac{c_{ls}}{2}\frac{h^+_i(x^+_{ils-1})^2 + h^-_i(x^-_{ils})^2}{x^+_{ils-1} + x^-_{ils}} + \frac{c_{ls}}{2}\frac{h^-_i(x^-_{ils-1})^2 + h^+_i(x^+_{ils})^2}{x^-_{ils-1} + x^+_{ils}}$$

Following the same steps as in *i)*,

$$J_{ils} - J'_{ils} < c_{ls}(h^+_i + h^-_i)\epsilon$$

which is a larger bound than in *i)* and *ii)*.

In all three cases, $\max(\epsilon, x^+_{ils-1} + x^-_{ils}) \geq x^+_{ils-1} + x^-_{ils}$ and $\max(\epsilon, x^-_{ils-1} + x^+_{ils}) \geq x^-_{ils-1} + x^+_{ils}$. Hence $J'_{ils} \leq J_{ils}$. □

Therefore, by taking ϵ small enough, the cost J is very well approximated by

$$J_{NLP} = \sum_{i=1}^{n} \sum_{l=1}^{T} \sum_{s=1}^{S_l} J'_{ils} \leq J$$

The number of variables in P^{ct} is $3n \sum_{l=1}^{T} S_l$ (u_{ils}, x^+_{ils}, and x^-_{ils} for n products, T periods, and S_l switching times per period). P^{ct}, with J_{NLP} as criterion, can only be solved for problems of small sizes with current algorithms, and optimality of the solution is not guaranteed. The nonlinearity in the criterion is introduced by Cases 3 and 4, in which the surplus sign changes during a period. In the next section, such events are forbidden by means of binary variables.

3.3 A Linear Programming (LP) model

Nonlinearities in J only appear in Cases 3 and 4. Hence, in order to have a linear objective, one should prevent the surplus from changing sign during a period, *i.e.*, to prevent the surplus going from inventory to backlog or from backlog to inventory, that is to ensure that $x_{ils-1} > 0 \Rightarrow x_{ils} \geq 0$, and $x_{ils-1} < 0 \Rightarrow x_{ils} \leq 0$. To do this, binary variables can be introduced and a mixed integer linear programming (MILP) can be derived (see [5]). In this formulation, because expression J^1_{ils} is equal to 0 in Case 2 and J^2_{ils} is equal to 0 in Case 1, the linear objective function can be written as $J^1_{ils} + J^2_{ils}$. However, even if the cost is now linear, the problem remains hard, because of binary variables, except for problems of small size.

To derive the LP model, it suffices to remove the binary variables in the MILP formulation, or equivalently ignore the nonlinear part in the exact cost J, *i.e.*,

$$P^{ct}_{LP} \begin{cases} \min J_{LP} = \sum_{i=1}^{n} \sum_{l=1}^{T} \sum_{s=1}^{S_l} \frac{c_{ls}}{2}[h^+_i(x^+_{ils-1} + x^+_{ils}) + h^-_i(x^-_{ils-1} + x^-_{ils})] & (16) \\ x^+_{ils} - x^-_{ils} = x^+_{ils-1} - x^-_{ils-1} + (u_{ils} - d_{il})c_{ls} \quad i = 1,..,n;\ l = 1,..,T; s = 1,..,S_l & (17) \\ \sum_{i}^{n} \tau_{ik} u_{ils} \leq 1 \quad \forall k \in M;\ l = 1,..,T; s = 1,..,S_l & (18) \\ u_{ils}, x^+_{ils}, x^-_{ils} \geq 0 \quad i = 1,..,n;\ l = 1,..,T; s = 1,..,S_l & (19) \end{cases}$$

In this model, the set of feasible rates is not reduced but, as shown in Proposition 2, the exact cost in Cases 3 and 4 given in Section 3.1 is approximate by a larger linear cost.

Proposition 2

Let u be a vector of feasible production rates and x the vector of surpluses induced by u. Let $J_{LP}(x)$ be the linear cost and $J(x)$ the exact cost (given in Section 3.1) given by the surplus vector x. Then,

$$J_{LP}(x) \geq J(x)$$

Proof:

Consider the four cases in Section 3.1. In Cases 1 and 2, J_{ils} is the same in J_{LP} and in J. However, in Case 3

$$J_{ils}(J) = \frac{c_{ls}}{2} \frac{h^+_i(x^+_{ils-1})^2 + h^-_i(x^-_{ils})^2}{x^+_{ils-1} + x^-_{ils}} = \frac{c_{ls}}{2} \left(\frac{h^+_i(x^+_{ils-1})^2}{x^+_{ils-1} + x^-_{ils}} + \frac{h^-_i(x^-_{ils})^2}{x^+_{ils-1} + x^-_{ils}} \right) \tag{20}$$

and

$$J_{ils}(J_{LP}) = \frac{h_i^+ c_{ls}(x^+_{ils-1} + x^+_{ils})}{2} + \frac{h_i^+ c_{ls}(x^-_{ils-1} + x^-_{ils})}{2} = \frac{c_{ls}}{2}(h_i^+ x^+_{ils-1} + h_i^- x^-_{ils}) \tag{21}$$

since $x^-_{ils-1} = x^+_{ils} = 0$.

Because the first term in $J_{ils}(J)$ is smaller than the first term in $J_{ils}(J_{LP})$, and the second term in $J_{ils}(J)$ is smaller than the second term in $J_{ils}(J_{LP})$, $J_{ils}(J) \leq J_{ils}(J_{LP})$. Symmetrically, the same is true in Case 4. □

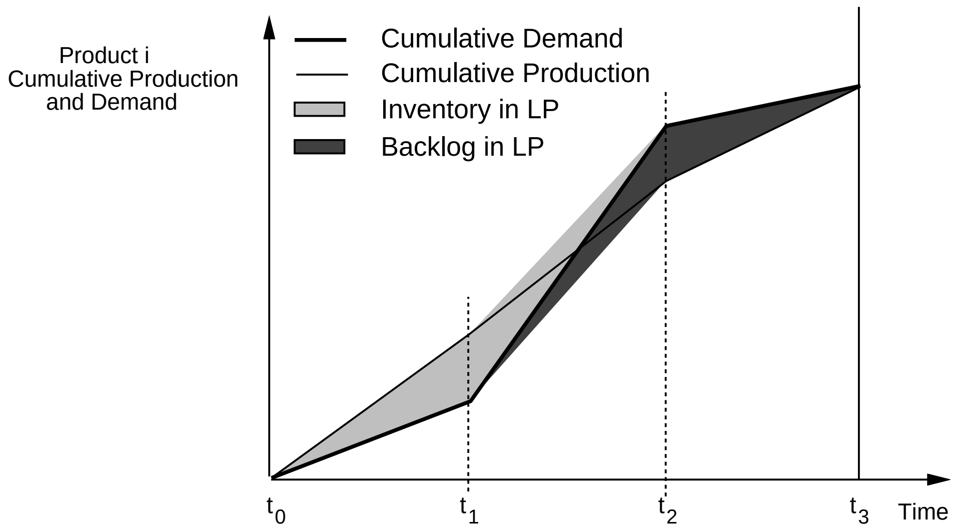

Figure 4: Inventory and backlog in the LP formulation

Hence, the optimal cost given by the LP model (J^*_{LP} and the associated surplus vector x^*_{LP}) is not exact, in the sense that it can differ from the exact cost $J(x^*_{LP})$. However, by Proposition 2 and as shown in Figure 4, J^*_{LP} is an upper bound of the exact cost. If the surplus is in an inventory state (*i.e.*, positive) at the beginning of a period and in a backlog state (negative) at the end ($[t_1, t_2]$ in Figure 4), the inventory cost in J_{LP} does not decrease after the crossing of the cumulative demand line, and behaves as if the surplus was equal to 0 at the end of the period. This can be seen in the inventory part of the LP cost. The first term of (21) does not depend on x^-_{ils}. Symmetrically, the backlog cost in J_{LP} does not depend on the value of the inventory at the beginning of the period (the second term of (21) does not depend on x^+_{ils-1}), and behaves as if the latter was equal to 0. Consequently, the LP model tends to avoid changes of the sign of the surplus in a period whenever possible.

It can also be shown that the LP model dominates the MILP model, since the exact cost of the optimal solution of the LP model is lower than or equal to the exact cost of the optimal solution of the MILP model (see [5]). This explains why the MILP formulation is not considered in the numerical experiments.

3.4 Comparing with a discrete time model

It is interesting to compare the LP model derived in the previous section to a more classical model in a discrete time approach (see [11] for instance).

$$P^{dt}\begin{cases}\min J_{dt} = \sum_{i=1}^{n}\sum_{l=0}^{T}\sum_{s=1}^{S_l}(h_i^+ c_{ls} x_{ils}^+ + h_i^- c_{ls} x_{ils}^-) & & (22)\\ x_{ils}^+ - x_{ils}^- = x_{ils-1}^+ - x_{ils-1}^- + U_{ils} - D_{ils} & i=1,..,n;\ l=1,..,T;\ s=1,..,S_l & (23)\\ \sum_{i=1}^{n}\tau_{ik}U_{ils} \leq c_{ls} & \forall k \in M;\ l=1,..,T;\ s=1,..,S_l & (24)\\ U_{ils}, x_{ils}^+, x_{ils}^- \geq 0 & i=1,..,n;\ l=1,..,T;\ s=1,..,S_l & (25)\end{cases}$$

where U_{ils} is the quantity of part type i produced between t_{ls} and t_{ls-1}, and D_{ils} is the demand quantity in i between t_{ls} and t_{ls-1}.

Because we consider that demand and production are constant between two switching times, rates and quantities are proportional, *i.e.*, $U_{ils} = c_{ls}u_{ils}$ and $D_{ils} = c_{ls}d_{il}$. Hence, the constraints are actually the same in P_{LP}^{ct} $((17) - (18))$ and P^{dt} $((23) - (24))$. Let us now compare their respective criteria J_{dt} and J_{LP}.

Remark 2

Let u be a vector of feasible production rates (or equivalently U a vector of feasible production quantities), and x or X be the vector of surpluses induced by u (or U). Let $J_{LP}(x)$ and $J_{dt}(X)$ be the costs given by the surplus vector. Then:

$$J_{dt}(X) = J_{LP}(x) + \sum_{i=1}^{n}\frac{c_{ls}}{2}(h_i^+ x_{i10}^+ + h_i^- x_{i10}^-) + \sum_{i=1}^{n}\frac{c_{ls}}{2}(h_i^+ x_{iTS_T}^+ + h_i^- x_{iTS_T}^-) \geq J_{LP}(x)$$

Both criteria are the same except for the cost at the first and last periods. Proposition 2 is verified when J_{LP} is replaced by J_{dt}. However, nothing can be said about the relationship between the exact costs of the optimal solutions x_{LP}^* and x_{dt}^*, *i.e.*, either $J(x_{LP}^*) \leq J(x_{dt}^*)$ or $J(x_{LP}^*) \geq J(x_{dt}^*)$. Broadly speaking, one can guess that, in particular on a long horizon, the solutions determined by solving P^{dt} and P_{LP}^{ct} will often be almost identical. This will even be more true for the first periods of the horizon, *i.e.*, often the only ones during which the production rates are actually implemented. However, one of the advantage of P_{LP}^{ct} compare to P^{dt} is that non-equal "periods" are handled in P_{LP}^{ct} in a more *natural* way.

The main interest of Section 3.3 lies in the propositions that were presented. In particular, Proposition 2 will help us in the following section to ensure that, by adding new switching times, **not only the linear cost will decrease but also the exact cost**.

4 Changing switching times

In this section, we present several propositions and a conjecture that will be used to derive the rules for adding and removing switching times, *given* the production rates and the surplus state, in the second step of the iterative procedure.

4.1 Some results

Let I denote the set of switching times $\{t_{11}, ..., t_{1S_1}, t_{21}, .., t_{TS_T}\}$, such that there is no $(t_{ls}, t_{l's'}) \in I \times I$ such that $t_{ls} = t_{l's'}$.

Proposition 3

Let P_{LP} (resp. P'_{LP}) be the linear programming model associated to the set of switching times I (resp. I'), and $I \subset I'$. Let $J_{LP}(x)$ be the linear cost of P_{LP} associated to the surplus vector x. Assume that x' is equal to x for every switching time in $I' \cap I$ and is chosen such that $u'_{ils'} = u_{ils}$ for every $t_{ls'}$ in $I' \setminus I$, where $t_{ls} \in I$ is such that $t_{ls-1} < t_{ls'} \leq t_{ls}$. Then

$$I \subset I' \Rightarrow J'_{LP}(x') \leq J_{LP}(x)$$

Proof: $I \subset I'$ implies that every switching time in I is also in I'. Let us choose a switching time $t_{ls'}$ such that $t_{ls'} \in I' \setminus I$, and $t_{ls-1} < t_{ls'} < t_{ls}$, $t_{ls} \in I$. Let $c^1_{ls} = t_{ls'} - t_{ls-1}$, and Let $c^2_{ls} = t_{ls} - t_{ls'}$, then $c_{ls} = c^1_{ls} + c^2_{ls}$. We want to show that the linear cost only decreases by adding $t_{ls'}$, and keeping the same production rates u_{ils} between t_{ls-1} and t_{ls}, *i.e.*:

$$c_{ls}/2[h^+_i(x^+_{ils-1} + x^+_{ils}) + h^-_i(x^-_{ils-1} + x^-_{ils})] \geq$$

$$c^1_{ls}/2[h^+_i(x^+_{ils-1} + x^+_{ils'}) + h^-_i(x^-_{ils-1} + x^-_{ils'})] + c^2_{ls}/2[h^+_i(x^+_{ils'} + x^+_{ils}) + h^-_i(x^-_{ils'} + x^-_{ils})] \qquad (26)$$

After replacing c_{ls} by $c^1_{ls} + c^2_{ls}$, and some simplifications,

$$c^1_{ls}(h^+_i x^+_{ils} + h^-_i x^-_{ils}) + c^2_{ls}(h^+_i x^+_{ils-1} + h^-_i x^-_{ils-1}) \geq c^1_{ls}(h^+_i x^+_{ils'} + h^-_i x^-_{ils'}) + c^2_{ls}(h^+_i x^+_{ils'} + h^-_i x^-_{ils'}) \quad (27)$$

Since either $x^-_{ils'} = 0$, or $x^+_{ils'} = 0$, let us suppose $x^-_{ils'} = 0$. Using Equation (17), we have:

$$x^+_{ils'} = x^+_{ils-1} - x^-_{ils-1} + c^1_{ls}(u_{ils} - d_{il}) \quad \text{and} \quad x^+_{ils'} = x^+_{ils} - x^-_{ils} - c^2_{ls}(u_{ils} - d_{il})$$

Plugging in (27) and, after simplification,

$$c^1_{ls}h^-_i x^-_{ils} + c^2_{ls}h^-_i x^-_{ils-1} + c^1_{ls}h^+_i x^-_{ils} + c^2_{ls}h^+_i x^-_{ils-1} \geq 0 \qquad (28)$$

which is satisfied since all the coefficients and the surplus variables are positive. Similarly, the same result can be obtained for $x^+_{ils'} = 0$.

After adding one by one all the switching times in $I' \setminus I$, with $u'_{ils'} = u_{ils} \;\; \forall (t_{ls'}, t_{ls}) \in I' \times I$ such that $t_{ls-1} < t_{ls'} \leq t_{ls}$, the resulting cost $J_{LP}(x') \leq J_{LP}(x)$. □

Corollary 1

Let J^*_{LP} be the optimal cost determined by solving P_{LP} with the set of switching times I, and J'^*_{LP} the optimal cost obtained by solving P'_{LP} with the set I'. Then,

$$I \subset I' \Rightarrow J'^*_{LP} \leq J^*_{LP}$$

This result means that the linear cost can only be improved by adding new switching times.

Proposition 4

Let $J(x^*)$ be the exact cost of an optimal solution determined with the set of switching times I, and $J(x'^*)$ the exact cost of an optimal solution determined with the set I'. If

$$\left| \begin{array}{ll} t_{ls'} = t_{ls-1} - \dfrac{x^+_{ils-1}}{u_{ils} - d_{il}} \in I' \quad \forall (i,l,s) \;\; \textit{such that} \;\; x^+_{ils-1}, x^-_{ils} > 0 & (29) \\ t_{ls'} = t_{ls-1} + \dfrac{x^-_{ils-1}}{u_{ils} - d_{il}} \in I' \quad \forall (i,l,s) \;\; \textit{such that} \;\; x^-_{ils-1}, x^+_{ils} > 0 & (30) \end{array} \right.$$

then,

$$I \subset I' \Rightarrow J(x'^*) \leq J(x^*) \;\; \forall x'^*, \; x^*$$

Proof:

With Equation (29), we have $x^+_{ils-1} + (u_{ils} - d_{il})(t_{ls'} - t_{ls-1}) = 0 = x_{ils'}$ (and the equivalent with Equation (30)).This means that we have a switching time at every crossing of the demands, then $J'_{LP}(x^*) = J(x^*)$, and with Proposition 2, $J(x'^*) \leq J'^*_{LP} \leq J'_{LP}(x^*) = J(x^*)$. □

Proposition 4 states that, by adding switching times at every crossing of the demand, the **exact cost can only be improved**.

Proposition 5

Let J^*_{LP} be the optimal linear cost determined with the set of switching times I. Let J'^*_{LP} be the optimal linear cost associated with the set I'. Let $t_{ls} \in I$ be a switching time not at the end of a period, i.e., $(l,s) \neq (l, S_l)\ \forall l$, *and let* $I' = I \backslash \{t_{ls}\} = \{t_{11}, .., t_{ls-1}, t_{ls+1}, .., t_{TS_T}\}$. *Then,*

$$J'^*_{LP} = J^*_{LP}$$

if and only if:

1. *All the production rates are the same before and after t_{ls}, i.e., $u^*_{ils} = u^*_{ils+1}\ \forall i$,*

2. *There is no crossing of the cumulative demand between t_{ls-1} and t_{ls+1}, i.e., $x^-_{ils-1} = x^-_{ils+1} = 0$ $\forall i$, or $x^+_{ils-1} = x^+_{ils+1} = 0\ \forall i$.*

Proof:

By Proposition 3, we know that $J'^*_{LP} \geq J^*_{LP}$.

Let us prove the *if* part. Condition 1 ensures $x'^+_{ils} - x'^-_{ils} = x^+_{ils} - x^-_{ils}$ by choosing $u'_{ils} = u^*_{ils}$ $\forall t_{ls} \in I$.

We want to show that Equation (26) — in which x_{ils+1} replaces x_{ils}, x_{ils} replaces $x_{ils'}$, c_{ls} replaces c^1_{ls}, c_{ls+1} replaces c^2_{ls}, and $c_{ls} + c_{ls+1}$ replaces c_{ls} — is an equality. Let us first consider $x^-_{ils-1} = x^-_{ils+1} = 0$, i.e., $x^-_{ils} = 0$ since $u^*_{ils} = u^*_{ils+1}$. After simplification, Equation (26) becomes:

$$c_{ls}x^+_{ils+1} + c_{ls+1}x^+_{ils-1} = c_{ls}x^+_{ils} + c_{ls+1}x^+_{ils}$$

which is verified by using Equation (17) as in the proof of Proposition 3. The same calculations can be performed for $x^+_{ils-1} = x^+_{ils+1} = 0$. Therefore $J'_{LP}(x') = J^*_{LP}$, and since $J'^*_{LP} \geq J^*_{LP}$, $J'^*_{LP} = J^*_{LP}$.

Now, let us prove the *only if* part. Since $J'^*_{LP} = J^*_{LP}$, then the optimal production rates u'^*_{ils} can also be used to find J^*_{LP} (see proof of Proposition 3), i.e., Condition 1 is satisfied. For these production rates, the equality between the costs can be once derived once again as in the proof of Proposition 3. When $x^-_{ils} = 0$, Equation (28) becomes $(h^+_i + h^-_i)(c_{ls}x^-_{ils+1} + c_{ls+1}x^-_{ils-1}) = 0$, which means that $x^-_{ils+1} = x^-_{ils-1} = 0$ since all the coefficients and surplus variables are positive. When $x^+_{ils} = 0$, the second condition in Condition 2 is derived. □

4.2 A conjecture

In this section, we present a conjecture based on the results for small problems with constant demand rates presented in [7]. When demand rates are constant, and supposing that there is enough capacity for each product to satisfy its demand, the goal is to reach the stationary state (zero surplus for every product) at minimum cost. One knows that the resulting optimal policy

– *i.e.*, the successive sets of optimal production rates – only depends on the surplus state. The surplus state space is actually divided into regions, and to each region is associated a vector of optimal production rates. This vector is always at a **corner** of the polyhedron induced by the capacity constraints on the machines and $u_{ils} \geq 0$, or at a corner of a *reduced polyhedron* created by fixing the production rates of some products to their respective demand rates (see Figure 5 for a two-part-type example).

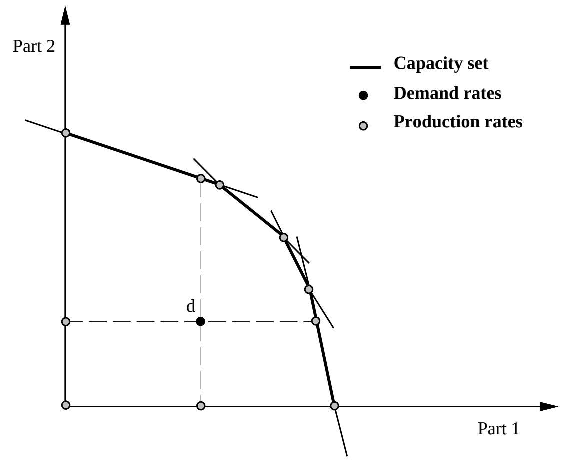

Figure 5: Corners of the original and reduced polyhedrons

With time-varying demand rates, the optimal policy does not only depend on the surplus state, but also on the time. Our conjecture is as follows.

Conjecture: *If a vector of production rates is optimal, then it is at a corner of a reduced polyhedron induced by the capacity constraints.*

Note that this is only a *necessary* condition, and that it might be the case that a vector at the corner of a reduced polyhedron is not optimal.

Our problem is now to be able to quickly check whether a vector of production rates satisfies the conjecture or not. The capacity constraints have the form $Au \leq b$ (Constraints (9)). We first need to find the set of active constraints $A_1 u = b_1$. Then, the production rates that are either equal to their associated demand rates or equal to zero are removed, and this becomes the system $A_1^{'} u' = b_1^{'}$. The vector of production rates is at the corner of the reduced polyhedron if the rank of $A_1^{'}$ (or equivalently A_1) is n', the dimension of u'. Actually, we can stop the calculation of the rank as soon as we know if it is n' or strictly lower. In the actual implementation, we consider that the capacity constraints are linearly independent, and we just check that the number of active constraints is larger than or equal to n'. If some constraints are linearly dependent, then the rank will be overestimated, and a vector might be considered to be at a corner even though it is not. This is not really a problem in the conjecture-based iterative procedure presented in Section 5.2, since it will not prevent interesting switching times from being added, and only some not interesting ones might be added.

5 The iterative procedure

Using Propositions 3, 4, and 5, several simple rules to create and remove switching times are derived, which ensure the cost – **both linear and exact** – to decrease. Figure 3 illustrates the iterative procedure. With fixed switching times, we optimize the production rates and obtain new values for u_{ils} and x_{ils}. Then, with fixed production rates, we use the rules to remove switching times and add new ones, and to obtain new values for S_l and c_{ls}. The first procedure is presented in Section 5.1. Using the conjecture, we present a new rule in Section 5.2 which introduces fewer switching times than the Rule-Based procedure. And fewer switching times introduced in the model implies a lower number of variables in P_{LP}.

5.1 A rule-based procedure

Based on the propositions presented in Section 4, let us introduce the following rules:

1. Remove every switching time that is not at the end of a period, for which the conditions in Proposition 5 are satisfied.

2. To each switching time t_{ls} after which the cumulative demand is crossed for a product i, add a new switching time in $([t_{ls}, t_{ls+1}]$, where the cumulative demand would be crossed if u_{ils} was used instead of u_{ils+1}.

3. To each switching time t_{ls} at which there is at least one product whose production rate changes, add a switching time at $t_{ls} - \varepsilon$ (if $t_{ls-1} < t_{ls} - \varepsilon$) and at $t_{ls} + \varepsilon$ (if $t_{ls+1} > t_{ls} + \varepsilon$).

4. When possible, set a new switching time at each time instant the cumulative demand is crossed.

The first rule allows the set of switching times to be reduced while ensuring, with Proposition 5, that the linear cost does not increase. Rule 2 is used to accelerate the procedure by guessing where the cumulative demands should be crossed. Rule 3 enables the production rates to change at different times than the ones originally fixed. The parameter ε corresponds to the variation authorized around a switching time. Hence, the number of iterations in the iterative procedure and the quality of the solution usually increase when ε decreases. In our numerical experiments, ε is chosen equal to 1. Finally, following Proposition 4, Rule 4 is used to guarantee that the exact cost is non-increasing when the LP model is solved again at the next step of the procedure. Actually, this rule ensures that the next linear cost is also lower than or equal to the previous exact cost.

Rule-Based (RB) procedure

- *Initialization. Choose S_l, $\forall l$ (the number of switching times per period), and c_{ls}, $\forall l, s$, the distance between the switching times. $J^0_{LP} = 0$. $k = 1$.*
- *Solve P^1_{LP}.*
- *While $(J^k_{LP} - J^{k-1}_{LP})/J^k_{LP} \geq \epsilon$ do*
  1. *$k = k + 1$. Apply Rules 1, 2, 3 and 4 in this order.*
  2. *Solve P^k_{LP} with the new set of switching times.*

Actually, Rule 3 can also be applied after Rules 2 and 4, and fewer switching times may be added in some cases.

5.2 A conjecture-based procedure

Rules 3 may introduce a large number of new switching times. We wish to introduce new switching times only when necessary, *i.e.*, only when the vector of production rates between two switching times does not satisfy our conjecture.

5. For each switching time t_{ls} such that the associated vector of production rates does not verify the conjecture, add a switching time at $(t_{ls-1}+t_{ls})/2$.

The Conjecture-Based (CB) procedure is the RB procedure in which Step 1 in the loop is replaced by:

1. $k = k+1$. *Apply Rules 1,5,2,4 in this order.*

One of the advantages of Rule 5 is that switching times are only added when required, thus reducing the number of switching times per iteration.

6 Reducing the number of switching times

The size of the linear programming problem is directly connected to the number of switching times. For each switching time t_{ls}, three variables are needed $(u_{ils}, x^+_{ils}, x^-_{ils})$ per part type. This is why reducing the total number of switching times is important.

In all our experiments, we noted that, at a switching time, production rates are often only changing for few part types, *i.e.*, a part type never requires exactly the same set of switching times as the other part types. Hence, by keeping only the necessary switching times, it is possible to create an *independent* set of switching times for every part type. However, less switching times means less flexibility since some of the interaction among part types is removed, and the resulting model might degrade the solution and give a larger cost. We shall see that this is not really the case in our numerical experiments, and that the computational benefits are significant.

Because there are different sets of switching times, the time intervals between production rate changes will differ from one part type to another. Let us denote by S_{il} the number of switching times for part type i in period l, and t_{ils} the s^{th} switching time of part type i in period l. S_l still denotes the overall number of switching times in period l. The linear programming model P^{ct}_{LP} becomes:

$$
P^{ct}_{LP,IS}\left\{
\begin{array}{llll}
\min J_{LP} = \sum_{i=1}^{n}\sum_{l=1}^{T}\sum_{s=1}^{S_{il}} \frac{c_{ls}}{2}[h^+_i(x^+_{ils-1}+x^+_{ils}) + h^-_i(x^-_{ils-1}+x^-_{ils})] & & & (31)\\
x^+_{ils} - x^-_{ils} & = x^+_{ils-1} - x^-_{ils-1} + (u_{ils}-d_{il})c_{ls} & i=1,..,n;\ l=1,..,T; s=1,..,S_{il} & (32)\\
\sum_{i}^{n} \tau_{ik} u_{ils'} & \leq 1 & \forall k \in M;\ l=1,..,T; s=1,..,S_l & \\
 & & \text{and } s' \text{ s.t. } t_{ils'} \leq t_{ls} \leq t_{ils'+1} & (33)\\
u_{ils}, x^+_{ils}, x^-_{ils} & \geq 0 & i=1,..,n;\ l=1,..,T; s=1,..,S_{il} & (34)
\end{array}\right.
$$

Even if every part type is treated individually, the capacity constraint has to take the production rates of all part types into account at every switching time. This is ensured through Constraint (33), where s' corresponds to the switching time of part type i associated to the production rate of i at time t_{ls}. On the other hand, Constraint (32), the inventory balance equation, has "only" to be satisfied for each part type individually. This would be different if part types were linked through a given bill of materials, *i.e.*, if some part types were components of others. In this case, production of a part type can only be started if the required components are available.

The rules presented in Section 5 need to be slightly adapted to manage independent sets of switching times. Rule 1 is now applied to every part type separately. Conditions 1 and 2 in Proposition 5 are for a given part type i and not for all part types i. Hence, differences between sets of switching times will already be created through Rule 1. Note that, as before and because of Condition 2, a switching time for a part type might be kept in the set even though its production rate does not change. Rules 2, 3 and 4 do not really change, except that every part type will also be considered separately, and thus the new switching times will clearly only be added to the set of the corresponding part type. Since Rule 5 is using the vector of production rates at a given switching time, a new switching time will be added after t_{ls} to every set of part types.

Iteration number	1	2	3	4	5
Common set of switching times					
Exact cost	**54474.04**	**52262.33**	**52119.03**	**52112.67**	**52112.67**
Variables in LP model	2010	2520	3270	3270	2370
CPU time (sec) - Total=21.23	2.29	4.84	4.74	4.59	4.77
Independent set of switching times					
Exact cost	**54474.04**	**52722.43**	**52319.69**	**52225.40**	**52225.40**
Variables in LP model	2010	327	357	387	387
CPU time (sec) - Total=6.56	2.29	0.52	1.02	1.94	0.79

Table 1: Comparing common and independent sets of switching times

The total number of variables in the linear program at each iteration of the procedure decreases drastically. Table 1 compares the conjecture-based procedure with a *common set of switching times* and with *independent sets of switching times* for a ten-product and eight-machine example. The number of iterations is the same in both cases, but the number of variables using independent sets of switching times is much smaller (the number of variables at the first iteration is the same because of the initialization step). The computation is much faster and the exact cost is not more than 0.2% over the exact cost obtained by using a common set of switching times. This illustrates that using the new formulation allows larger problems to be treated.

7 Numerical experiments

NLP problems are solved using GAMS version 2.05 [3], a package for large and complex mathematical programming problems, using a *reduced-gradient* algorithm combined with a *quasi-Newton* algorithm, when nonlinearities only appear in the objective function. CPlex version 4.0.7 [13], a very efficient commercial software, has been used to solve the LP problems.

7.1 Example 1

In this example, the production of four part types has to be planned in a three-machine workshop. The planning horizon has four periods of 100 time units each ($c_l = 100 \;\; \forall l$). Data of the problem are given in Table 2. Costs are the same for each product: $h_i^+ = 10$, $h_i^- = 100 \;\; \forall i$.

In our experiments, the number of switching times $S-1$ within a period was fixed and identical for each period. In a period, switching times were chosen to be equidistant, *i.e.*, $c_{ls} = c_l/S \;\; \forall l, s$. Figure 6 illustrates the results obtained when $S = 10$ (a switching time every ten time units) in

	Processing time on machine			Demand rate in period				Initial
	1	2	3	1	2	3	4	surplus
Product 1	0.2	0.1	-	1	2	1	1	100
Product 2	0.3	0.2	0.2	1	1	1	0	-100
Product 3	0.1	0.1	0.2	2	2	1	2	-100
Product 4	0.1	0.2	0.1	4	2	1	5	100

Table 2: Example 1

the LP case. Since each production curve is near its demand curve, the demand curves are not explicitly identified. At the end of the horizon, only product 4 is in a backlog state.

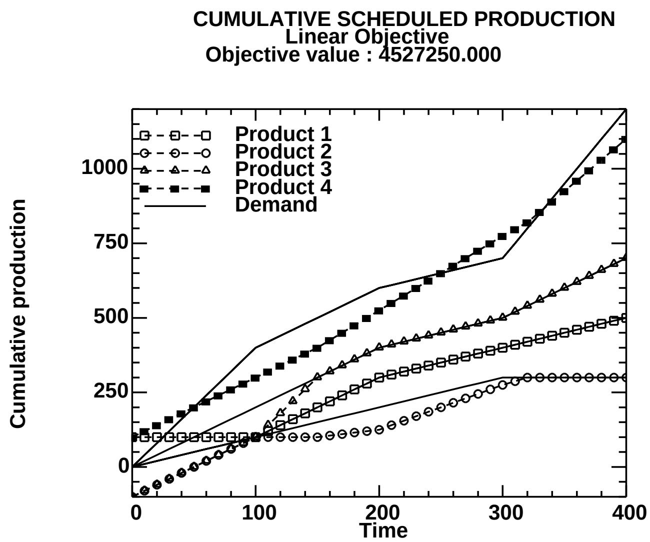

Figure 6: Example 1, $S = 10$

Final costs obtained with the NLP and LP formulation are given and compared in Table 3. When $S = 33$, the problem is too large for the nonlinear objective function. Consequently, it cannot be solved with GAMS.

Adding switching times improves the cost in both cases, but the improvement is more significant in the LP model. As the number of switching times increases, *i.e.*, as the length of the time intervals decreases, the difference between the optimal linear and nonlinear costs becomes negligible. Also, the linear cost, given in Table 3, is an upper bound of the exact cost, as shown in Proposition 2. Thus, the gain by using the NLP formulation is an upper bound of the actual gain. This remark shows that, besides its practical interest (see CPU times in Table 3), the LP formulation can be nearly as accurate as the NLP provided it has enough switching times, or *well chosen* switching times (near crossing of the demands).

Looking at the cost obtained in Table 3, in the LP model, from $S = 4$ to $S = 5$, one can see that it has increased (also true from $S = 20$ to $S = 33$). Switching times can be found every 25 time units (c_l/S) in the first case, and every 20 time units in the second, *i.e.*, they are different within each period. This example illustrates the fact that it is more important to enable production rates to switch at the right times than often.

This is what the iterative procedure is doing. Its performance is analyzed in more detail in the example of Section 7.3. Table 4 shows the results given by using the conjecture-based procedure

Number S of switching times per period	$S=1$	$S=2$	$S=4$	$S=5$	$S=10$	$S=20$	$S=33$
Linear Programming Formulation							
CPU Time (Sec)	0.06	0.06	0.15	0.17	0.54	1.48	3.46
Objective Value	**5350000**	**4612500**	**4543750**	**4557000**	**4527250**	**4525875**	**4526125**
Gain compare to $S=1$ (in %)	-	13.79	15.07	14.82	15.38	15.40	15.40
Nonlinear Programming Formulation							
CPU Time (Sec)	10	15	25	30	70	260	-
Objective Value	**4706794**	**4660937**	**4532291**	**4530416**	**4525416**	**4525416**	-
Gain compare to $S=1$ (in %)	-	0.97	3.71	3.75	3.85	3.85	-
Gain with NLP Formul. (in %)	12.02	-1.05	0.25	0.58	0.04	0.01	-

Table 3: Comparison between LP and NLP models

with independent sets of switching times, where costs of the LP formulation in Table 3 are the costs at the first iteration of the procedure. Whatever the initial number of switching times, the final cost is always close to the best one given by the NLP formulation (at most 0.004 % larger). In Table 4, when the initial number of switching times is greater than 5, the CPU time start to increase. This could be explained by the size of the LP model which is bigger and takes a long time to be loaded.

Initial number S of switching times per period	$S=1$	$S=2$	$S=4$	$S=5$	$S=10$	$S=20$	$S=33$
Initial cost	5350000	4612500	4543750	4557000	4527250	4525875	4526125
Final cost	4525495	4525513	4525578	4525416	4525416	4525416	4525442
Number of iterations	13	11	4	3	3	3	3
CPU Time (Sec)	1.15	0.94	0.42	0.44	0.69	1.67	3.81

Table 4: Using the iterative procedure

7.2 Example 2

In our second experiment, we consider the one-machine case with multiple part types and constant demands. We only consider stable problems, in which demand rates are feasible, *i.e.*, the surplus state eventually attains and remains at 0 for every product. Connolly [4] showed that, in the two-part-type case, the optimal policy is to give priority to the product with the greatest ratio of backlog cost to processing time (h_i^-/τ_i). All the capacity of the machine is devoted to this product (maximum production rate) if it is in a backlog state, and the production rate is equal to the demand rate as soon as the surplus reaches zero. At that time, the remaining capacity is devoted to the other product until its surplus also becomes zero. This result is extended for the multiple-part-type case in [10]. Products are ranked in the order of their ratio, and priority is given from the highest to the lowest ratio.

In an example with four products, the backlog cost and the demand were the same for each product, and the processing time was smaller for product i than for product $i+1$ (so $h_i^-/\tau_i \geq h_{i+1}^-/\tau_{i+1}$). There were two periods of 100 time units. Figure 7 presents the behavior of the production rates with nine switching times within each period ($S=10$, *i.e.*, a switching time every ten time units). Between $t=0$ and $t=10$, all the capacity is devoted to product 1. However, between $t=10$ and $t=20$, the capacity of the machine is shared between product 1 and product 2. This is because product 1 does not need all the capacity to reach a zero surplus state, and some capacity is available for product 2, the product with the second highest ratio.

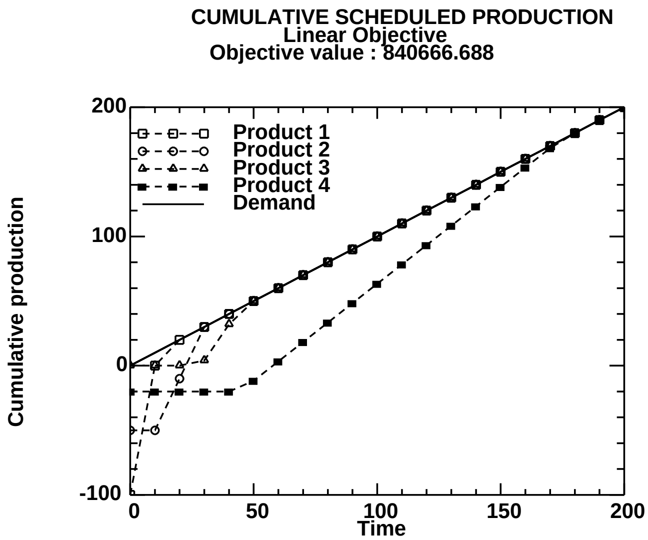

Figure 7: Example 2, $S = 10$

Again, we used the iterative procedure to determine the necessary switching times. The solution is shown in Figure 8. The priority between products appears more clearly, since no part type i is produced before part types 1 to $i-1$ have reached a zero surplus state.

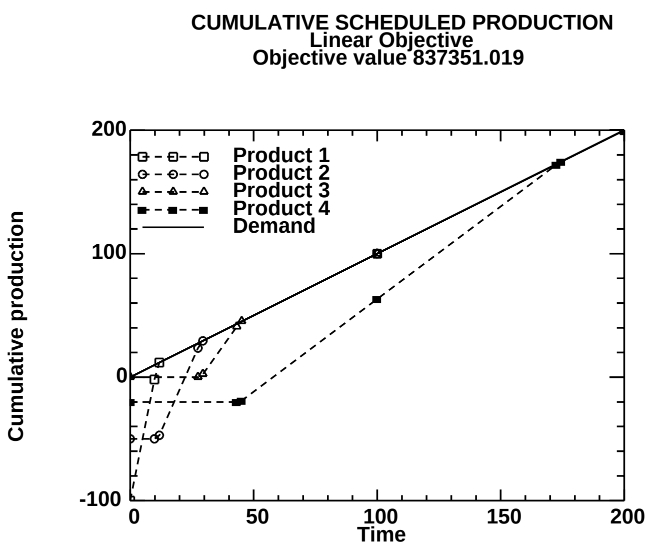

Figure 8: Example 2, after iterative procedure

7.3 Example 3

In the following examples, the horizon has 6 periods of 100 time units each ($c_l = 100, \ \forall l$). The demand is piecewise constant for both.

Comparison between the two iterative procedures

Table 5 presents a comparison between the rule-based procedure and the conjecture-based procedure for a ten-product and eight-machine example using a common set of switching times.

As in the previous examples, in Table 5, $S = 16$ means that there are 15 switching times within each period, 97 switching times for the whole horizon ($97 = 6 \times 15 + 6 + 1$, 15 switching times in each period, one switching time for the beginning of each period and one switching time for the end of the last period) at the start of the procedure and only 45 switching times remain at the end. The improvement of the exact cost is 1.94% after 6 iterations.

In Table 5, the number of iterations is smaller in the conjecture-based procedure than in the rule-based procedure. The conjecture-based procedure only adds switching times when necessary (see section 4.2). The final number of switching times is nearly the same in both procedures, but the average number of switching times is smaller in the conjecture-based procedure.

The exact cost computed by the conjecture-based procedure in the worst case ($S = 1$) is 1.9% from the exact cost in the rule-based procedure with $S = 1$. In the best case, the exact cost determined by the conjecture-based procedure only differs by 0.01% from the one determined by the rule-based procedure.

Initial number of switching times per period	$S = 1$	$S = 6$	$S = 11$	$S = 16$	$S = 21$
Rule-based procedure					
Exact cost	**52471.64**	**52108.48**	**52152.13**	**52120.86**	**52107.86**
Final S	45	43	44	45	44
Improvement (%)	74.20	10.88	4.26	1.94	1.04
Number of iterations	100 (max)	17	9	6	5
Average S	77.3	88.7	94.6	101.5	109.4
Conjecture-based procedure					
Exact cost	**53471.92**	**52867.12**	**52166.20**	**52159.74**	**52116.58**
Final S	34	39	46	43	46
Improvement (%)	73.89	9.52	4.24	1.87	1.02
Number of iterations	8	6	6	4	4
Average S	22.8	35.3	50.5	57.2	71.5

Table 5: A ten-product and eight-machine example

Comparison with a discrete time formulation

The problem could also be solved by discretizing the horizon in very short periods of equal lengths, and using the discrete time formulation P^{dt}. A good approximation of P^{dt} is obtained by setting a switching time at every time unit, or every two time units in the LP model P^{ct}_{LP}. The underestimation of J_{dt} induced by using J_{LP} is small enough to be ignored.

Table 6 compares the solution of the discrete time formulation with a switching time every time unit (Initial number of swt. per period = 100), and two time units (Initial number of swt. per period = 50), and two runs of the conjecture-based procedure with 21 and 11 switching times per period at the initialization step. The CPU time row is the CPU time in seconds for each iteration. Table 7 displays the results of a twenty-product, eight-machine example. One can see that the total CPU times of the iterative procedure are much smaller than the ones of the discrete time formulation, and the exact costs are at most 0.24% larger than the best cost obtained in

Discrete-time formulation					
Initial number of swt. per period	100	50			
Exact cost	**52110.62**	**52156.89**			
CPU time (sec)	259	55			
Variables in LP model	18030	9030			
Continuous-time formulation					
Iteration number	1	2	3	4	5
Initial number of swt. per period	21				
Exact cost	52636.40	52210.66	52151.34	**52151.26**	
CPU time (sec) - Total=11.50	7.54	1.13	2.02	0.81	
Variables in LP model	3810	378	426	426	
Initial number of swt. per period	11				
Exact cost	54441.60	52658.83	52319.06	52236.81	**52236.81**
CPU time (sec) - Total=6.37	2.02	0.52	1.02	2.00	0.81
Variables in LP model	2010	327	357	387	387

Table 6: Discrete and continous time formulations - Ten products

Table 6, and 0.12% in Table 7. Moreover, the costs given by the iterative procedure are similar in Table 6, and smaller in Table 7 than the cost obtained by using the discrete time formulation in which a switching time is set every two time units.

Discrete-time formulation					
Initial number of swt. per period	100	50			
Exact cost	**108319.20**	**108486.01**			
CPU time (sec)	699	139			
Variables in LP model	36060	18060			
Continuous-time formulation					
Iteration number	1	2	3	4	5
Initial number of swt. per period	21				
Exact cost	109876.25	108628.43	108453.71	**108453.71**	
CPU time (sec) - Total=35.58	17.50	4.81	7.73	5.54	
Variables in LP model	7620	822	885	888	
Initial number of swt. per period	11				
Exact cost	113742.22	109551.92	108683.46	108419.40	**108417.93**
CPU time (sec) - Total=16.72	3.85	1.38	2.44	5.16	3.89
Variables in LP model	4020	651	762	783	777

Table 7: Discrete time model and continous time model - Twenty products

8 Conclusion

This paper provides an efficient two-step iterative procedure for solving a continuous-time production planning problem with piecewise constant demand and production rates. Three models were presented to optimize production rates given switching times, *i.e.*, times at which production rates are allowed to change. Properties of these models have been discussed and, because of its computational efficiency, the linear programming model is used at the first step of the procedure. Various rules have been derived to change the set of switching times in the second step. Two versions of the iterative procedure are actually proposed, the rule-based and conjecture-based procedures. We have also shown that the procedure could be considerably speed up by managing different sets of switching times for each product.

Numerical experiments support the fact that the conjecture-based procedure is more efficient than the rule-based procedure in terms of computing times, and is nearly as cost-effective. The same thing can be said about using independent sets of switching times compared to a common set.

Our future work is organized around three main goals: making the iterative procedure more efficient, enlarging its scope, and using it to derive new results. The procedure might be improved by finding new rules and new combinations of existing rules. As in Sharifnia [9], buffers between machines and setup times are two important things that need to be considered in the future. Finally, the procedure could be used to determine the optimal control policy in some special cases, in particular when demand rates are constant, to find the best trajectory from an initial state (inventory and backlog for some products) to the steady state, which in our case corresponds to a zero-surplus level for each product. For instance, they can be helpful in the hedging-point policies in the hierarchical model presented in [6], when surpluses move towards a hedging point in the time period between two uncontrollable events (see example in Section 7.2). The components of a hedging point are the inventory levels to attain in order to anticipate future failures of the system. In the same vein, and when buffers are taken into account, the optimal draining of fluid re-entrant lines might be tackled using our approach (see [12]).

References

[1] Billington, P.J., Mcclain, J.O. and Thomas, L.J. (1983) Mathematical Programming Approaches to Capacity-Constrained MRP systems: Review, Formulation and Problem Reduction. *Management Science*, **29**, 1126-1141.

[2] Bitran, G.R. and Tirupati, D. (1993) Hierarchical Production Planning. Graves, S.C., Rinnooy Kan, A.H.G., and Zipkin, P.H. (eds.), *Logistics of Production and Inventory*, Handbooks in Operations Research and Management Science, **4**, 523-568.

[3] Brooke, A., Kendrick, D. and Meeraus, A. (1988) *GAMS: A User's Guide*, The Scientific Press.

[4] Connolly, S. (1992) A Real-time Policy for Performing Setup Changes in a Manufacturing System. *MIT Operations Research Center, Master's Thesis, n⁰ LMP-92-005*, Massachusetts Institute of Technology.

[5] Dauzere-Peres, S., and Gershwin, S.B., "Continuous-Time Planning Models", *MIT Laboratory for Manufacturing and Productivity, Report n⁰ LMP-93-007*, Massachusetts Institute of Technology, Cambridge, Massachusetts.

[6] Gershwin, S.B. (1989) Hierarchical Flow Control: A Framework for Scheduling and Planning Discrete Events in Manufacturing Systems. *Proceedings of the IEEE 77*, 195-209.

[7] Gershwin, S.B. (1994) *Manufacturing Systems Engineering*, Prentice Hall.

[8] Hachman, S.H. and Leachman, R.C. (1989) A General Framework for Modeling Production. *Management Science* **35**, 478-495.

AJOUTER LA REF KHMELNITSKY SI ARRIVEE A TEMPS

[9] Sharifnia, A. (1994) Stability and Performance of Distributed Production Control Methods based on Continuous Flow Models. *IEEE Transactions on Automatic Control* **39** (4), 725-737.

[10] Srivatsan, N. (1993) Synthesis of Optimal Policies for Stochastic Manufacturing Systems, *MIT Operations Research Center, PhD thesis, n^0 LMP-93-009*, Massachusetts Institute of Technology.

[11] Thomas, L.J. and McClain, J.O. (1993) An Overview of Production Planning. Graves, S.C., Rinnooy Kan, A.H.G., and Zipkin, P.H. (eds.), *Logistics of Production and Inventory*, Handbooks in Operations Research and Management Science, **4**, 333-370.

[12] Weiss, G. (1996) Optimal Draining of Fluid Re-Entrant Lines: Some Solved Examples. Kelly, F.P., Zachary, S. and Ziedins, I. (eds.), Stochastic Networks: Theory and Applications, 19-34.

[13] Using the CPLEX Callable Library (1995) CPLEX Optimization Inc.