Journal of Global Optimization **16:** 271–294, 2000.

Optimal Scheduling in Parallel and Serial Manufacturing Systems via the Maximum Principle

KONSTANTIN KOGAN[1] and TOSHIHIDE IBARAKI[2]
[1]*Tel-Aviv University, Department of Industrial Engineering, Tel-Aviv, Israel*
(e-mail: kogan@wine.cteh.ac.it)
[b]*Kyoto University, Department of Applied Mathematics and Physics, Kyoto, Japan*

(Received 9 April 1997; accepted in revised form 24 October 1999)

Abstract. The problems of M-machine, J-product, N-time point preemptive scheduling in parallel and serial production systems are the focus of this paper. The objective is to minimize the sum of the costs related to inventory level and production rate along a planning horizon. Although the problem is NP-hard, the application of the maximum principle reduces it into a well-tractable type of the two-point boundary value problem. As a result, algorithms of $O(NMJ^{k(N-L)+1})$ and $O(N(MJ)^{k(N-L)+1})$ time complexities are developed for parallel and serial production systems, respectively, where L is the time point when the demand starts and k is the ratio of backlog cost over the inventory cost. This compares favorably with the time complexity $O((J+1)^{MN})$ of a naive enumeration algorithm.

Key words: Optimization; Production scheduling; The maximum principle

1. Introduction

So far numerous efforts have been undertaken for effective production scheduling throughout available facilities to achieve a goal of economic and, thus, competitive manufacturing. Unfortunately, only very special cases of the multi-machine scheduling problems can be solved in polynomial time (see tectbooks like Brucker, 1995; Lawler et al., 1993; Pinedo, 1995) while the rest is *NP*-hard. Therefore, the increases in the numbers of machines and products lead to combinatorial explosion and the problem cannot be solved for practical purposes.

Since production is a dynamic phenomenon, dynamic representation appears to be a natural way to model it, especially when dealing with demands for product types, which change in time, i.e., dynamic lot-sizing. While the dynamic and mixed-integer programming approaches (Crowston and Wagner, 1973; Karmarkar et al., 1987) are directly applicable to handle such problems, these approaches usually require exponential computational time. On the other hand, we will show in this paper that the optimal control theory turns out to be an efficient tool in the search for well-solvable cases.

Kimemia and Gershwin (1983) first presented manufacturing as a continuous-time controllable product flow that passes through workstations (where setups are negligible) and buffers. They described this dynamic model by differential equations. The near-optimal flow was found from the linear problem formulated at required moments of time by varying a cost functional. Later, Sousa and Pereira (1992), Khmelnitsky et al. (1995), Kogan et al. (1997), applied the maximum principle to derive projected gradient-based methods for different dynamic scheduling problems. These methods are characterized by the polynomial computational time when a relatively rough accuracy is required. However, the more accurate solution is needed the worse convergence is observed, and the computation time becomes exponential.

An alternative in studying a production flow control problem with the aid of the maximum principle is to reduce it to a two-point boundary-value problem for ordinary differential equations. Then, standard methods as, for example, the shooting can be applied to solve this problem (Khmelnitsky and Kogan, 1994). However, as is the case with classical combinatorial techniques, such methods can realistically handle problems with only few machines.

The present paper suggests an approach that can take advantage of both the analytical characteristics obtained of the maximum principle and the numerical accuracy of the combinatorics. In this approach, shooting becomes more efficient, and our algorithm is able to cope with sizable number of machines and products, in two typical production environments: parallel and serial manufacturing systems.

2. Problem formulations

We consider a set of product types $\{j| = 1, \ldots, J\}$ and a set of machines $\{m|m = 1, \ldots, M\}$ capable of processing those product types. Every machine m is characterized by its fixed capacity P_{mj} for processing product type j units per time unit. A control variable $p_{mj}(t)$ is introduced to model the fact that, at any point of time t, machine m is either processing job j with constant production rate P_{mj} or it is idle:

$$p_{mj}(t) = \begin{cases} P_{mj} \text{ if machine } m \text{ is processing job } j \text{ at time } t \\ 0, \text{ otherwise .} \end{cases} \tag{1}$$

In this section we present continuous-time scheduling formulations which are more general for the dynamic problems under consideration. In the subsequent sections, these formulations are transformed into the conventional discrete forms for which our algorithm will be applied. We present formulations for parallel and serial production systems separately.

2.1. PARALLEL PRODUCTION SYSTEM

To formalize a parallel production system, we introduce inventory level $X_j(t)$ of product type j at time t, as the flow of this product type through the parallel machines. More exactly, the rate of change in the inventory level $X_j(t)$ is the difference between the total production rate of machines currently producing j and the current demand for j:

$$\dot{X}_j(t) = \sum_m p_{mj}(t) - d_j(t) , \quad X_j(0) = X_j^0 , \ j = 1, \dots, J , \tag{2}$$

where $d_j(t)$ is the number of units of product type j per time unit required at moment t (demand rate) and X_j^0 is the initial inventory level of product type j.

Note, that in this formulation, classical time parameters such as processing times and due dates are stated in more general notion of rates. To obtain classical formulations of scheduling problems, the differential equation (2) is to be discretized, i.e., to be replaced with its difference form while demands are set to zero along the overall planning horizon T except the time points specified by the due dates. Specifically, if $X_j^0 = 0$ and $d_j(t) = 1$ for t equal to the due date of product type j and $d_j(t) = 0$ otherwise, we obtain a model for preemptive scheduling of J products through M parallel machines.

According to Equation (2), every machine is capable of processing several different products simultaneously. If it is not the case, the following constraint ensures that only one product type is processed on a machine at a time:

$$\sum_j \frac{p_{mj}(t)}{P_{mj}} \leqslant 1 , \quad m = 1, \dots, M . \tag{3}$$

The objective of our problem is to minimize the following cost with respect to the inventory levels and production rates:

$$\min \int_0^T \left(\sum_j C(X_j(t)) + \sum_m \sum_j S(p_{mj}(t)) \right) \mathrm{d}t , \tag{4}$$

subject to constraints (1)–(3).

The cost $C(X_j(t))$ represents inventory holding costs if $X_j(t) > 0$ (surplus) and backlogging costs if $X_j(t) < 0$ (shortage). If the model (1)–(4) is applied for preemptive scheduling, these costs become dynamic penalties for earliness and tardiness, respectively. The second term, $S(p_{mj}(t))$, represents the cost of producing product type j on machine m. Although we consider some special forms of $C(X_j(t))$ in Section 5, the functional form of $S(p_{mj}(t))$ will remain arbitrary throughout the paper. The latter is of importance, because as is often the case in industry, the actual cost of producing on a machine cannot be represented by a linear or convex function. Hereafter, we shall call $C(X_j(t))$ as the inventory cost function and $S(p_{mj}(t))$ as the production cost function.

2.2. SERIAL PRODUCTION SYSTEM

To formalize the model for a serial production system, we have to take into account intermediate buffers for every product type. The difference of parallel and serial production systems is illustrated in Figure 1. The inventory or buffer level $X_{mj}(t)$ of product type j at time t is also related to machines m by which product type j has

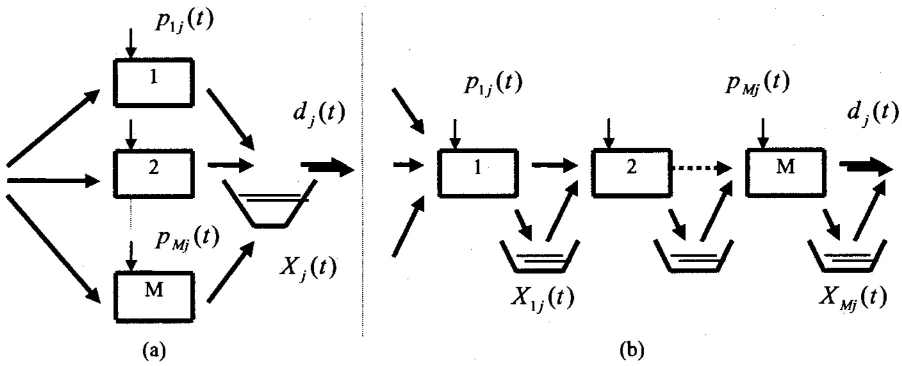

Figure 1. Parallel (a) and serial (b) production systems.

been processed. As a result, the inventory flow of a product type through an intermediate buffer is determined by the difference between the current production rates of two consecutive machines that process product type j:

$$\begin{aligned} &\dot{X}_{mj}(t) = p_{mj}(t) - d_j(t)\,, \quad j = 1, \ldots, J,\ m = M \\ &\dot{X}_{mj}(t) = p_{mj}(t) - p_{m+1j}(t)\,, \quad j = 1, \ldots, J,\ m = 1, \ldots, M-1 \\ &X_{mj}(0) = X^0_{mj}\,, \quad j = 1, \ldots, J,\ m = 1, \ldots, M\,. \end{aligned} \tag{5}$$

Constraint (3) remains without change, but the scheduling objective in the serial system is slightly modified to minimize work-in-process in all buffers:

$$\min \int_0^T \sum_m \sum_j (C(X_{mj}(t)) + S(p_{mj}(t)))\, \mathrm{d}t\,. \tag{6}$$

3. Discrete time formulations and their complexity

To handle conventional scheduling problems in our formulations, constraints and objective functions (1)–(6) are converted into the form of discrete times (mesh points) t_n which are equally spaced throughout the planning horizon:

$$t_0 = 0\,, \quad t_{n+1} - t_n = \Delta(period)\,, \quad n = 0, 1, \ldots, N-1\,,\ t_N = T\,.$$

3.1. PARALLEL PRODUCTION SYSTEM

$$\text{minimize} \sum_n \sum_j C(X_j(t_n)) + \sum_n \sum_m \sum_j S(p_{mj}(t_n)) \tag{7}$$

subject to

$$X_j(t_{n+1}) - X_j(t_n) = \Big(\sum_m p_{mj}(t_n) - d_j(t_n)\Big)\,\Delta, \quad X_j(t_0) = X^0_j\,,$$
$$j = 1, \ldots, J,\ n = 0, \ldots, N-1 \tag{8}$$

$$\sum_j \frac{p_{mj}(t_n)}{P_{mj}} \leqslant 1\,, \quad m = 1, \ldots, M, \quad n = 0, \ldots, N-1\,. \tag{9}$$

3.2. SERIAL PRODUCTION SYSTEM

$$\text{minimize} \sum_n \sum_m \sum_j \left(C(X_{mj}(t_n)) + S(p_{mj}(t_n))\right) \tag{10}$$

subject to

$$\begin{aligned} &X_{mj}(t_{n+1}) - X_{mj}(t_n) = (p_{mj}(t_n) - d_j(t_n))\,\Delta\,, \\ &\qquad\qquad m = M,\ j = 1, \ldots, J,\ n = 0, \ldots, N-1 \\ &X_{mj}(t_{n+1}) - X_{mj}(t_n) = (p_{mj}(t_n) - p_{m+1j}(t_n))\,\Delta\,, \\ &\qquad\qquad m = 1, \ldots, M-1,\ \ j = 1, \ldots, J, n = 0, \ldots, N-1 \end{aligned} \tag{11}$$

$$\sum_j \frac{p_{mj}(t_n)}{P_{mj}} \leqslant 1\,, \quad m = 1, \ldots, M,\ n = 0, \ldots, N-1\,. \tag{12}$$

The scheduling problems (7)–(9) and (10)–(12) are *NP*-hard if we consider the manufacturing system in which machines have different capability. Since the proof is similar for both parallel and serial systems, it is presented here by reducing problem T3P to scheduling of only a parallel manufacturing system (SPM) with inventory cost function $C(X_j(t)) = kc_j|X_j(t)|$ if $X_j(t) < 0$ and $C(X_j(t)) = c_j|X_j(t)|$ otherwise. T3P is a variant of the well-known *NP*-complete problem 3P (3-partition). The proof of T3P *NP*-hardness is quite technical and, therefore, relegated to the appendix.

3.2.1. *T3P* (*triple 3-partition*)

Instance: A finite set A of $9p$ elements, a bound $B \in Z^+$ and a size $s(a) \in Z^+$ for each $a \in A$, such that each $s(a)$ satisfies $B/4 < s(a) < B/2$, such that $\Sigma_{a \in A}\, s(a) = 3pB$ and such that, for each $a \in A$, there are two other elements $a', a'' \in A$ for which $s(a) = s(a') = s(a'')$.

Question: Can A be partitioned into $3p$ disjoint sets $S_1, S_2, \ldots, S_{3p}$, such that $\Sigma_{a \in S_j}\, s(a) = B$ for $j = 1, 2, \ldots, 3p$. (Note, that each S_j contains exactly three elements.)

THEOREM 1. *SPM is NP-hard.*

Proof. In order to reduce T3P to SPM, we consider the following instance of SPM corresponding to a given instance of T3P, where we assume without loss of generality that: $A = \{a_1, a_2, \ldots, a_{3p}, \ldots, a_{9p}\}$ and $s(a_m) = s(a_{m+3p}) = s(a_{m+6p}) = P_m$, $m = 1, 2, \ldots, 3p$, where $P_m \in Z^+$ (positive integers).

Instance of SPM: There are $M = 3p$ machines and $J = 3p$ jobs (i.e., $M = J$ holds). Production rate P_{mj} of machine m for processing job j is defined by:

$$P_{mj} = P_m\,, \quad m = 1, \ldots, M \ (P_{mj} \text{ is independent of } j)\,.$$

We consider discrete time $t = t_n = 0, 1, 2, \ldots, n = 0, \ldots, N$ $(\Delta = 1)$ and consider that machine m can produce P_m number of products of job j in each unit time interval. The demand of job j is set to:

$$d_t(t) = \begin{cases} B, & \text{if } t = 3 \\ 0, & \text{otherwise}\,. \end{cases}$$

We set constant k of cost function $C(X_j(t)) = kc_j|X_j(t)|$ for $X_j(t) < 0$ large enough. Then, it is not difficult to see that the schedule of completing all jobs before its deadline $t = 3$ is better than any other schedule that processes some jobs after $t = 3$. Indeed, if we assume

$$C(X_j(t)) = \begin{cases} |X_j(t)|\,, & t \leqslant 3 \\ k|X_j(t)|\,, & t > 3\,, \end{cases}$$

the total cost of any schedule that completes all jobs before their deadline $t = 3$ is

$$\sum_t \sum_j C(X_j(t)) = \Big(\sum_m P_m\Big)(2+1) = 3\sum_m P_m = 3pB\,.$$

If there is any job that is processed after the deadline, then the cost is at least k. Therefore, if we set $k > 3pB$, then a schedule is optimal if and only if there is no job processed after the deadline.

Now we claim that the above instance of SPM has a schedule that completes all jobs before their deadline $t = 3$, if and only if the given instance of T3P has a solution. It is immediate to see that SPM is *NP*-hard if this claim holds.

First, assuming that there is a schedule that processes all jobs before their deadline, we show that there is a solution to T3P. Let job j be processed at machine j_1, j_2 and j_3 at time intervals [0, 1], [1, 2] and [2, 3], respectively. Then, since job j meets its deadline, we have

$$P_{j_1} + P_{j_2} + P_{j_3} \geqslant B\,.$$

Since this holds for all $j = 1, 2, \ldots, 3p$, and it is assumed $3\sum_m P_m = 3pB$, the above must hold by equality:

$$P_{j_1} + P_{j_2} + P_{j_3} = B, \text{ for all } j\,.$$

Therefore, this gives a solution to T3P.

To prove the converse, assume that there is a solution $(S_1, S_2 \ldots, S_{3_p})$ to T3P. If $S_j = (P_{j_1} + P_{j_2} + P_{j_3})$, where $1 \leqslant j_1, j_2, j_3 \leqslant 3p$, we consider the schedule that assigns job j to machines j_1, j_2, j_3 for the three time intervals in some order. We have to show that this can actually define a schedule by appropriately ordering j_1, j_2, j_3 for each job j (corresponding to time intervals [0, 1], [1, 2], [2, 3]), so that each machine m processes exactly one job at each time interval and no job is processed on different machines at the same time interval. This is accomplished by constructing a schedule for intervals [0, 1], [1, 2], [2, 3], separately in this order. For interval [0, 1], we construct the bipartite graph $G_1 = (V_1, V_2, E)$ as follows:

$$V_1 = \{1, 2 \ldots, M\}, \quad V_2 = \{1, 2 \ldots, M\}$$
$$(m, j) \in E \Leftrightarrow P_m \in S_j .$$

Also, we consider the weight $w(m, j)$ for $(m, j) \in E$ as the number of occurrences P_m in S_j (recall that the same P_m may appear more than once in S_j). Then G_1 satisfies:

$$\sum_{j|(m,j)\in E} w(m, j) = 3 \text{ for } m = 1, 2, \ldots, M$$
$$\sum_{m|(m,j)\in E} w(m, j) = 3 \text{ for } j = 1, 2, \ldots, M .$$

Let $Z(m) = \{j \in V_2 | (m, j) \in E\}$. Then for any subset $U \subseteq V_1$ we have:

$$\left| \bigcup_{m\in U} Z(m) \right| \geqslant |U| \tag{13}$$

because

$$\sum_{m\in U,(m,j)\in E} w(m, j) = \sum_{m\in U} \sum_{j|(m,j)\in E} w(m, j) = 3|U|$$
$$= \sum_{j\in Z(U),(m,j)\in E,m\in U} w(m, j) \leqslant 3|Z(U)| ,$$

where $Z(U) = \cup_{m\in U} Z(m)$ (the inequality holds because

$$\sum_{(m,j)\in E,m\in U} w(m, j) \leqslant \sum_{(m,j)\in E} w(m, j) = 3$$

for all j). The relation (13) is known as the complete marriage condition (Ahuja et al., 1993) and tells that graph G_1 has a complete matching $M_1 \subseteq E$, i.e., $|M_1| = M$ and, for each $m \in V_1$ there is exactly one $(m, j) \in M_1$, and, for each $j \in V_2$, there is exactly one $(m, j) \in M_1$. We consider that this matching M_1 defines the schedule for time interval [0, 1], i.e., assigns job j to machine m in time interval [0, 1] if and only if $(m, j) \in M_1$.

For time interval [1, 2], we first delete from each S_j the element P_m chosen by $(m, j) \in M_1$. For the resulting $S_1, S_2, \ldots, S_M$, we construct bipartite graph $G_2 = (V_1, V_2, E)$. This G_2 satisfies:

$$\sum_{j|(m,j)\in E} w(m, j) = 2 \text{ for } m = 1, 2, \ldots, M$$
$$\sum_{m|(m,j)\in E} w(m, j) = 2 \text{ for } j = 1, 2, \ldots, M .$$

Hence, by similar argument, we can show that (13) holds for G_2, and there is a complete matching $M_2 \subseteq E$. Then define the schedule for time interval [1, 2] by this M_2.

Finally, for the last time interval [2, 3], the argument is similar, except that

$$\sum_{j|(m,j)\in E} w(m, j) = 2 \text{ for } m = 1, 2, \dots, M$$
$$\sum_{m|(m,j)\in E} w(m, j) = 1 \text{ for } j = 1, 2, \dots, M$$

holds in this case.

This completes the proof for the converse direction, and hence the proof of the claim. □

4. Necessary optimality conditions

Although the maximum principle is equally stated for both continuous and discrete time systems, the analytical investigation of their behavior is naturally carried out for the general, continuous-time formulations, first, and then is transformed to discrete time formulations by time decomposition technique.

Both problems (1)–(6) are stated in the continuous-time canonical form of the optimal control; i.e., to optimize the objective presented as an integral of the system variables under the differential equations for continuous state variables $X_j(t)/X_{mj}(t)$ with initial boundary conditions, and the constraints for measurable bounded control variables $p_{mj}(t)$. The maximum principle applied to such a system asserts that there exist left-continuous functions of bounded variation (dual variables) $\psi_j(t)/\psi_{mj}(t)$ so that the following dual differential equations and transversality conditions hold (stated separately for parallel and serial production systems).

4.1. PARALLEL PRODUCTION SYSTEM

$$\dot{\psi}_j(t) = \frac{\partial C(X_j(t))}{\partial X_j}, \quad \psi_j(T) = 0, \ j = 1, \dots, J. \tag{14}$$

4.2. SERIAL PRODUCTION SYSTEM

$$\dot{\psi}_{mj}(t) = \frac{\partial C(X_{mj}(t))}{\partial X_{mj}}, \quad \psi_{mj}(T) = 0, \ j = 1, \dots, J, \ m = 1, \dots, M. \tag{15}$$

The maximum principle says that the optimal control strategy is achieved by maximizing the following function $H(t)$, called Hamiltonian, under constraints (1) and (3).

4.3. PARALLEL PRODUCTION SYSTEM

$$H(t) = -\sum_j C(X_j(t)) - \sum_m \sum_j S(p_{mj}(t)) + \sum_j \psi_j(t)\Big(\sum_m p_{mj}(t) - d_j(t)\Big) . \qquad (16)$$

4.4. SERIAL PRODUCTION SYSTEM

$$H(t) = -\sum_m \sum_j (C(X_{mj}(t)) + S(p_{mj}(t))) + \sum_{m=1}^{M-1} \sum_j \psi_{mj}(t)(p_{mj}(t) - p_{m+1j}(t)) + \sum_j \psi_{Mj}(t)(p_{Mj}(t) - d_j(t)) . \qquad (17)$$

LEMMA 1. *Given the problems for the parallel production system (1)–(4), and (1), (3), (5), (6) for the serial production system, optimal production rate* $p_{mj}(t)$ *satisfies the following conditions (necessary condition for optimality) described in three types of regimes via the dual variables* $\psi_j(t)$ *and* $\psi_{mj}(t)$.

(i) PRODUCTION REGIME

Parallel production system

$$p_{mj}(t) = P_{mj},\ p_{mj'}(t) = 0 ,$$
$$\text{if } P_{mj}\psi_j(t) - S(P_{mj}) > 0 \text{ and } P_{mj}\psi_j(t) - S(P_{mj}) > P_{mj'}\psi_{j'}(t) - S(P_{mj'}),$$
$$\forall j' \neq j, \forall m .$$

Serial production system

$$p_{mj}(t) = P_{mj},\ p_{mj'}(t) = 0 ,$$
$$\text{if } P_{mj}(\psi_{mj}(t) - \psi_{m-1j}(t)) - S(P_{mj}) > 0 \text{ and}$$
$$P_{mj}(\psi_{mj}(t) - \psi_{m-1j}(t)) - S(P_{mj}) > P_{mj'}(\psi_{mj'}(t) - \psi_{m-1j'}(t)) - S(P_{mj'}),$$
$$\forall j' \neq j,\ \forall m \neq 1 ;$$
$$P_{mj}\psi_{mj}(t) - S(P_{mj}) > 0 \text{ and } P_{mj}\psi_{mj}(t) - S(P_{mj}) > P_{mj'}\psi_{mj'}(t) - S(P_{mj'}),$$
$$\forall j' \neq j, m = 1 .$$

(ii) NO-PRODUCTION REGIME

Parallel production system

$$p_{mj}(t) = 0 ,$$
$$\text{if } P_{mj}\psi_j(t) - S(P_{mj}) < 0 \quad \forall j, m .$$

Serial production system

$$p_{mj}(t) = 0, \text{ if } P_{mj}(\psi_{mj}(t) - \psi_{m-1j}(t)) - S(P_{mj}) < 0 \quad \forall j, m \neq 1 ;$$
$$P_{mj}\psi_{mj}(t) - S(P_{mj}) < 0 \quad \forall j,\ m = 1 .$$

(iii) SINGULAR REGIMES

Parallel production system

$$p_{mj}(t) \in \{0, P_{mj}\},\ p_{mj''}(t) = 0\ ,$$
$$\text{if } P_{mj}\psi_j(t) - S(P_{mj}) = 0 \text{ and } P_{mj}\psi_{j''}(t) - S(P_{mj''}) < 0, \quad \forall m,\ \forall j'' \neq j.$$
$$p_{mj}(t) \in \{0, P_{mj}\},\ p_{mj'}(t) \in \{0, P_{mj'}\},\ p_{mj''}(t) = 0\ ,$$
$$\text{if } P_{mj}\psi_j(t) - S(P_{mj}) = P_{mj'}\psi_{j'}(t) - S(P_{mj'}) > P_{mj''}\psi_{j''}(t) - S(P_{mj''})\ ,$$
$$\forall m,\ \forall j' \neq j'' \neq j\ .$$

Serial production system

$$p_{mj}(t) \in \{0, P_{mj}\},\ p_{mj''}(t) = 0\ ,$$
$$\textit{if } P_{mj}(\psi_{mj}(t) - \psi_{m-1j}(t)) - S(P_{mj}) = 0 \text{ and } P_{mj''}(t) - \psi_{m-1j''}(t)) - S(P_{mj''}) < 0\ ,$$
$$\forall m \neq 1,\ \forall j'' \neq j\ ;$$
$$P_{mj}\psi_{mj}(t) - S(P_{mj}) = 0, \text{ and } P_{mj''}\psi_{mj''}(t) - S(P_{mj''}) < 0\ , \quad m = 1,\ \forall j'' \neq j\ .$$
$$p_{mj}(t) \in \{0, P_{mj}\},\ p_{mj'}(t) \in \{0, P_{mj'}\},\ p_{mj''}(t) = 0\ ,$$
$$\text{if } P_{mj}(\psi_{mj}(t) - \psi_{m-1j}(t)) - S(P_{mj}) = P_{mj'}(\psi_{mj'}(t) - \psi_{m-1j'}(t)) - S(P_{mj'})$$
$$> P_{mj''}(\psi_{mj''}(t) - \psi_{m-1j''}(t)) - S(P_{mj''})\ ,$$
$$\forall m \neq 1,\ \forall j' \neq j'' \neq j\ ;$$
$$P_{mj}\psi_{mj}(t) - S(P_{mj}) = P_{mj'}\psi_{mj'}(t) - S(P_{mj'}) > P_{mj''}\psi_{mj''}(t) - S(P_{mj''})\ ,$$
$$m = 1,\ \forall j' \neq j'' \neq j\ .$$

Proof. Since there is no constraint that relates production rates of different machines, maximization of the Hamiltonian is separated to individual machines:

Parallel production system

$$H_m(t) = \sum_j \left[\psi_j(t)p_{mj}(t) - S(p_{mj}(t)\right]\ , \quad m = 1, \ldots, M\ . \tag{18}$$

Serial production system

$$\begin{aligned} H_m(t) &= \sum_j \left[\psi_{mj}(t)p_{mj}(t) - S(p_{mj}(t))\right]\ , \quad m = 1\ ; \\ H_m(t) &= \sum_j \left[(\psi_{mj}(t) - \psi_{m-1j}(t))p_{mj}(t) - S(p_{mj}(t))\right]\ , \quad m = 2, 3, \ldots, M\ . \end{aligned} \tag{19}$$

The maximum principle implies that the optimal controls $p_{mj}(t)$ are obtained by maximizing the Hamiltonians (18) and (19) at every point of time t. Since at every point of time the dual variables $\psi_j(t)/\psi_{mj}(t)$ are constants and every machine is allowed to produce only one product type at a time (constraint (3)), the maximum of the Hamiltonian (18)/(19) is defined by J comparisons of its values. These values are calculated by setting production of machine m on every product type j, i.e., the corresponding control variable is set to its maximal value, while for all the other product types the machine controls are set at zero. If the maximal value of the Hamiltonian found on all J control combinations is unique and positive we have determined an optimal production regime, in case it is negative there will be no production. Finally, if the maximum is either zero or not unique, singular regimes with uncertain controls appear on the optimal trajectory as stated in this lemma. □

Note, that the singular regimes (iii) represent the points of time where it is optimal for a machine to produce either a number of products simultaneously with a given rate or with much lesser rate than given. The former models a situation when there is a number of urgent orders that demand high inventories thereby causing machine chattering (chattering regime). The latter (maximum of the Hamiltonian equals to zero) models a situation when there are either no orders except a low demand for only one product or a number of orders, but they are insignificant in comparison to the system capacity (workless regime). Both regimes require exact equalities of the Hamiltonian maximum involving different production rates and different types of cost functions and, therefore, are very unlikely to occur.

If singular regimes can be avoided everywhere or almost everywhere on the optimal trajectory, then Lemma 1 along with equations (2), (14) and (5), (15) give two two-point boundary-value problems for the optimality of the two corresponding dynamic scheduling problems for parallel and serial production systems. Such two-point boundary-value problems are commonly solved by guessing initial values for the dual variables, integrating both primal and dual systems of differential equations in the same direction (from left to right), and then comparing the obtained and given terminal values for the dual variables to correct the guess. Evidently, this shooting is of exponential nature and applicable only to very small manufacturing systems. In what follows, by a special choice of the inventory cost functions, we derive a well-tractable case for both production systems.

5. Production systems with special inventory costs

In this section we specialize the inventory cost function as follows:

$$C(X_j(t)) = \begin{cases} c_j|X_j(t)|, & \text{for } X_j(t) \geqslant 0 \\ kc_j|X_j(t)|, & \text{for } X_j(t) < 0\,, \end{cases} \tag{20}$$

$$C(X_{mj}(t)) = \begin{cases} c_{mj}|X_{mj}(t)|, & \text{for } X_{mj}(t) \geqslant 0 \\ kc_{mj}|X_{mj}(t)|, & \text{for } X_{mj}(t) < 0\,, \end{cases} \tag{21}$$

where k is a positive integer number. This cost $C(X)$ was selected for proving NP-hardness of our problems in Theorem 1 and it is illustrated in Figure 2. For this cost function we show that the dual variables $\psi_j(t)$ and $\psi_{mj}(t)$ have finite upper and

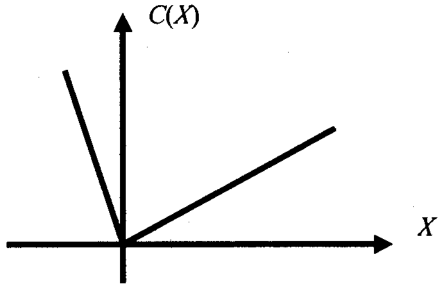

Figure 2. Specialized inventory cost function.

lower bounds for their initial values $\psi_j(0)$ and $\psi_{mj}(0)$ respectively (Lemma 2). Moreover, it will be shown in the next section, that for the discrete-time models with the specialized inventory cost function, the set of values, $\psi_j(0)$ and $\psi_{mj}(0)$ can take on, is also bounded (Lemma 3).

LEMMA 2. *Consider the dual problems, i.e., maximizing the Hamiltonian (16) under (1), (3), (14) (for the parallel production system) and maximizing the Hamiltonian (17) under (1), (3), (15) (for the serial production system). If there exists a* $t \in [0, T]$ *such that* $X_j(\tau) \neq 0$, $X_{mj}(\tau) \neq 0$, *for all* $\tau \in [t, T]$, *then* $\psi_j(0)$ *and* $\psi_{mj}(0)$ *are bounded as follows*:

$$-c_j T \leqslant \psi_j(0) \leqslant kc_j T \text{ and } -c_{mj} T \leqslant \psi_{mj}(0) \leqslant kc_{mj} T . \tag{22}$$

Proof. Let us consider the dual differential equations (15) for the serial production system. Substituting derivatives of the cost function (21) (which exist over the interval $[t, T]$) into equations (15), we find:

$$\begin{aligned} \psi_{mj}(t) &= \psi_{mj}(T) - \int_t^T c_{mj}\, \mathrm{d}\tau, \text{ if } X_{mj}(\tau) > 0 , \text{ for all } \tau \in [t, T] \\ \psi_{mj}(t) &= \psi_{mj}(T) + \int_t^T kc_{mj}\, \mathrm{d}\tau, \text{ if } X_{mj}(\tau) < 0 , \text{ for all } \tau \in [t, T] . \end{aligned} \tag{23}$$

Taking into account the terminal boundary condition of Equation (15), and considering two extreme cases, namely, inventory values are either only positive or only negative along the planning horizon (i.e., only one product from several demanded is produced along the entire planning horizon) we immediately obtain inequality (22) for the initial values of the serial system dual variables.

Similarly we can obtain inequality (22) for the parallel production system. □

6. Complexity to solve discrete-time models

To further clarify the choice of the inventory related cost functions, we return to the discrete formulation of the problems presented in Section 3, and calculate values of the dual variables from the corresponding dual difference equations.

LEMMA 3. *Given dual problems (1), (3), (14), (16) (for the parallel production system) and (1), (3), (15), (17) (for the serial production system) with inventory cost functions (19) and (20), respectively (but the production cost functions* $S(p_{mj}(t))$ *in (16) and (17) are still arbitrary), if*

$$X_j(t_n) \neq 0\,, \quad X_{mj}(t_n) \neq 0\,, \ \textit{for} \ n = 0, 1, \ldots, N-1\,, \tag{24}$$

then

$$\begin{aligned} &-c_j N\Delta \leqslant \psi_j(0) = r_j c_j \Delta \leqslant k c_j N\Delta \ \textit{and} \\ &-c_{mj} N\Delta \leqslant \psi_{mj}(0) = r_{mj} c_{mj} \Delta \leqslant k c_{mj} N\Delta \end{aligned} \tag{25}$$

hold for some integers r_j, r_{mj}

$$r_j, r_{mj} \in \{-N, -N+1, -N+2, \ldots, 0, 1, 2, \ldots, kN-2, kN-1, kN\},$$

respectively.

Proof. Since for the discrete problem formulations (7) and (10), the dual equations (14) and (15) take also difference forms (Bryson and Ho, 1969), Equations (22) in Lemma 2 for the serial production system are transformed as follows:

$$\begin{aligned} &\psi_{mj}(t_n) = \psi_{mj}(t_{n+1}) - c_{mj}\Delta\,, \ \text{if} \ X_{mj}(t_n) > 0 \\ &\psi_{mj}(t_n) = \psi_{mj}(t_{n+1}) + k c_{mj}\Delta\,, \ \text{if} \ X_{mj}(t_n) < 0, \end{aligned} \tag{26}$$

while the terminal boundary conditions for the dual variables remain the same:

$$\psi_{mj}(t_N) = \psi_{mj}(T) = 0\,.$$

Next, by starting from the terminal boundary conditions, considering the two extreme cases presented in Lemma 2 (i.e., $X_{mj}(t_n)$ are either always positive or always negative along the planning horizon) and calculating recursively by Equations (26) all values of the dual variables we obtain bounds for $\psi_{mj}(0)$ stated in condition (25) for the descrete problem formulations. Moreover, from the described recursive procedure it immediately follows that initial values of the dual variables (as well as evidently any non-terminal their values) can take on only multiples of $c_{mj}\Delta$ no matter whether $X_{mj}(t_n)$ change their sign at any points t_n, $n = 0, 1, \ldots, N-1$ of the planning horizon or not. This completes the proof of condition (25) for the serial production system.

The equations for the dual parallel production system become

$$\begin{aligned} &\psi_j(t_n) = \psi_j(t_{n+1}) - c_j\Delta\,, \ \text{when} \ X_j(t_n) > 0 \\ &\psi_j(t_n) = \psi_j(t_{n+1}) + k c_j\Delta\,, \ \text{when} \ X_j(t_n) < 0 \end{aligned} \tag{27}$$

The proof is similar to the case of serial production. □

From Lemma 3 follows that there are generally

$$J^{N(k+1)+1} \text{ and } (JM)^{N(k+1)+1}$$

possible initial values of the dual variables in parallel and serial systems, respectively. However, not all combinations appear to be legitimate when demands meet some realistic restrictions. The next lemma elaborates on this by considering a point t_L with $0 \leqslant L \leqslant N-2$, at which the first demand for some product is set in the system. The possibility, $L = N-1$ is excluded from consideration because, in this case no tardiness is possible. To present such a situation, additional inventory constraint must be introduced into the models to prohibit the tardiness. This, in turn, change the terminal boundary constraints for the corresponding dual variables and, thus, makes all proven lemmas illegitimate.

LEMMA 4. *Given primal problems (7)–(9) and (10)–(12) in parallel and serial production systems with inventory cost functions (20) and (21) (but, production cost function $S(p_{mj}(t))$ remains arbitrary), respectively, if $X_j(t_n) \neq 0$, $X_{mj}(t_n) \neq 0$, hold for all $n = 0, 1, \ldots, N-1$, and $X_j(t_0) \geqslant \xi$, $X_{mj}(t_0) \geqslant \xi$ hold for an infinitesimal $\xi > 0$, and L is such that for all j, $d_j(t_n) = 0$ for $n = 0, 1, \ldots, L$, and $d_j(t_L)\Delta > X_j(t_0)$ for some j, then the initial values of dual variables $\psi_j(0)$ and $\psi_{mj}(0)$ of the corresponding dual problems are multiples of $c_j\Delta$ and $c_{mj}\Delta$, respectively*:

$$\begin{aligned} &c_j\Delta(1-L) \leqslant \psi_j(0) \leqslant (k(N-L)-L)c_j\Delta \\ &c_{mj}\Delta(1-L) \leqslant \psi_{mj}(0) \leqslant (k(N-L)-L)c_{mj}\Delta\,. \end{aligned} \tag{28}$$

Proof. To prove that $\psi_j(0)$ and $\psi_{mj}(0)$ are multiples of $c_j\Delta$ and $c_{mj}\Delta$, respectively, it is sufficient to note that all conditions of Lemma 3 satisfied in Lemma 4.

To prove the bounds (28), consider a product type j for which a demand is set at t_L in the parallel production system. Since the initial inventory level of every product is given as a positive value and the demand for the product j starts not earlier than at point t_L, the corresponding dual variable can only grow linearly over the interval $[0, t_L]$ (see dual equations (27) in Lemma 3). As a result, at point t_L it can become equal at least to its minimal positive value, i.e., to $c_j\Delta$.

This is due to the fact that any non-positive value of a dual variable means no production (see regime (ii), Lemma 1), i.e., at point t_L such that $d_j(t_L)\Delta > X_j(t_0)$, the demand will make inventories negative, and hence, result in decrease of the dual variable after point t_L up to the end of the planning horizon, which implies that this dual variable will remain negative and the required zero terminal condition $\psi_j(t_N) = 0$ will never be realized.

At the same time, a positive value of the dual variable (at least the minimal one) at point t_L makes possible production (see regime (i), Lemma 1), at the subsequent period despite negative inventories, which decrease the dual variable to a negative value. This production can be sufficient to compensate the negative inventories at points succeeding t_L and, thus, increase back the dual variable at the minimal rate c_j to meet the terminal condition $\psi_j(t_N) = 0$.

On the other hand, the maximal positive value that the dual variable is able to attain by the linear growth is $kc_j\Delta(N - L)$, which is due to the maximal rate kc_j available for getting into the same zero terminal condition in the $N - L$ remaining periods.

Thus, we obtain simple equations for determining the upper bound for the initial values of the dual variables:

$$\psi_j(0) + Lc\Delta = kc_j\Delta(N - L)\ ;$$

as well as the equations for their lower bound:

$$\psi_j(0) + Lc_j\Delta = c_j\Delta\ .$$

Condition (28) for the parallel production systems immediately results from these equations. These are evidently the worst case estimation, because not all product demands necessarily start at L.

The equations for determining the upper and lower bounds for the dual serial production system become

$$\psi_{mj}(0) + Lc_{mj}\Delta = kc_{mj}\Delta(N - L)\ ;$$
$$\psi_{mj}(0) + Lc_{mj}\Delta = c_{mj}\Delta\ .$$

The proof is similar to the case of parallel production. □

As we have seen so far, the maximum principle delivers necessary optimality conditions for both continuous-time and discrete formulations of the parallel and serial production systems. In the following theorem, sufficient conditions are given for only discrete formulations, which make use of initial values of dual variables predetermined in Lemmas 3 or 4.

Theorem 2. *Consider primal problems (7)–(9) (for the parallel production system) and (10)–(12) (for the serial production system) with inventory cost functions (19) and (20) and arbitrary production cost function $S(p_{mj}(t))$, respectively.*

Parallel production system

If inequality (24) is satisfied and the following condition holds:

$$r_jP_{mj}c_j - S(P_{mj}) \neq r_{j'}P_{mj'}c_{j'} - S(P_{mj'})\ ,$$
$$\text{for all } t_j, r_{j'} \in \{-N, -N+1, \ldots, -1, 0, 1, \ldots, kN-1, kN\}$$
$$\text{and all } m,\ j \neq j'\ ; \tag{29}$$
$$r_jP_{mj}c_j - S(P_{mj}) \neq 0, \text{ for all } r_j \in \{-N, -N+1, \ldots, -1, 0, 1, \ldots, kN-1, kN\}$$
$$\text{and all } m,\ j,$$

then the global optimal solution $X_j(t_n)$ and $\psi_j(t_n)$ is obtained as:

$$\arg\min_{\{\psi_j(0)\}} \sum_n \sum_j c_j |X_j(t_n)| + \sum_n \sum_m \sum_j S(p_{mj}(t_n)) , \tag{30}$$

where $X_j(t_n)$ are the solutions of Equations (7) and $\psi_j(t_n)$ are the solutions of Equations (27) both computed from left to right with the controls $p_{mj}(t_n)$ determined by Lemma 1. The initial values of the dual variables $\psi_j(0)$ are chosen within the bounds (25) so that the terminal condition $\psi_j(T) = 0$ is met.

Serial production system

If inequality (24) is satisfied and the following condition holds:

$$\begin{aligned}
&r_j P_{mj} c_j - S(P_{mj}) \neq r_{j'} P_{mj'} c_{j'} - S(P_{mj'}) ,\\
&\text{for all } r_j, r_{j'} \in \{-N, -N+1, \ldots, -1, 0, 1, \ldots, kN-1, kN\} ,\\
&\quad m = 1 \text{ and all } j \neq j' ;\\
&P_{mj}(r_j c_{mj} - r'_j c_{m-1j}) - S(P_{mj}) \neq P_{mj'}(r_{j'} c_{mj'} - r'_{j'} c_{m-1j'}) - S(P_{mj'}) ,\\
&\text{for all } r_j, r'_j, r_{j'}, r'_{j'} \in \{-N, -N+1, \ldots, -1, 0, 1, \ldots, kN-1, kN\}\\
&\quad \text{and all } m \neq 1,\ j \neq j' ;\\
&P_{mj}(r_j c_{mj} - r'_j c_{m-1j}) - S(P_{mj}) \neq 0 ,\\
&\text{for all } r_j, r'_j \in \{-N, -N+1, \ldots, -1, 0, 1, \ldots, kN-1, kN\} \text{ and all } j, m \neq 1 ;\\
&r_j P_{mj} c_j - S(P_{mj}) \neq 0, \text{ for all } r_j \in \{-N, -N+1, \ldots, -1, 0, 1, \ldots, kN-1, kN\} ,\\
&\quad m = 1 \text{ and all } j ,
\end{aligned} \tag{31}$$

then the global optimal solution $X_{mj}(t_n)$ and $\psi_{mj}(t_n)$ is obtained as:

$$\arg\min_{\{\psi_{mj}(0)\}} \sum_n \sum_m \sum_j (c_{mj} |X_{mj}(t_n)| + S(p_{mj}(t))) , \tag{32}$$

where $X_{mj}(t_n)$ are the solutions of Equations (10) and $\psi_{mj}(t_n)$ are the solutions of Equations (26), both computed from left to right with the controls $p_{mj}(t_n)$ determined by Lemma 1. The initial values of the dual variables $\psi_{mj}(0)$ are chosen within the bounds (25) so that the terminal condition $\psi_{mj}(T) = 0$ is met.

Proof. Let us consider only the parallel production system. It follows from the previous discussion that an optimal solution of the problem (7)–(9) satisfies the system of the primal (7) and dual (27) equations as well as the terminal boundary condition $\psi_j(T) = 0$, where the control variables $p_{mj}(t)$ are determined by Lemma 1. This is however only a necessary condition, since the maximum principle can guarantee only local optimality.

In case there is a number of local optimal solutions, the solution minimizing the objective (30) among them is the global optimal solution. Moreover, if inventories are not zero at the mesh points of time as stated in condition (24), it is sufficient to solve the primal and dual equations for the fixed number of initial boundary values determined by (25) (see Lemma 3).

However, in order for the controls to be unambiguous when computing this solution and, thus, the proof complete, there must not be singular regimes (iii) (see Lemma 1) on the optimal trajectory. By replacing the dual variables in conditions (iii) of Lemma 1 with their values determined in Lemma 3, the remaining condition (29) which excludes the singular regimes (iii) is readily obtained. □

REMARK. Theorem 2 presents a straightforward scheduling algorithm of $O(J^{N(k+1)+1})$ (for the serial production system J is replaced with multiplication JM to yield $O((JM)^{N(k+1)+1})$) iterations corresponding to all initial values of $\psi_j(t)$ and $\psi_{mj}(t)$ within the bounds (25). According to Lemma 1, every iteration requires $O(NMJ)$ time. The number of iterations can be reduced if demands are concentrated at $N-L \geqslant 2$ points from the end of the planning horizon and there exists product j such that $d_j(t_L)\Delta$ exceeds the initial inventories: $d_j(t_L)\Delta > X_j(t_0) \geqslant \xi$ ($\xi > 0$ is an infinitesimal). By Lemma 4, the worst case estimate of the number of iterations, in this case becomes $O(J^{k(N-L)})$ and $O((MJ)^{k(N-L)})$ for the parallel and serial systems, respectively and the overall complexity of the algorithm becomes $O(NMJ^{k(N-L)+1})$ for the parallel production system and $O(N(MJ)^{k(N-L)+1})$ for the serial production system, respectively. Thus, for example, cubic running time is expected for a special case with $L=N-2$ and $k=1$. Although the general complexity is not polynomial in N, it still compares favorably with the time complexity $O((J+1)^{MN})$ of a naive enumeration algorithm of all schedules. This improvement becomes possible because our approach enumerates only the initial values of dual variables.

7. Algorithm and illustrative example

As noted in the remark to Theorem 2, the following algorithm (described for both parallel and serial production systems) can obtain optimal solutions:

Step 1. Select initial values of the dual variables from the set defined by condition (25) or (28) of Lemma 3 or Lemma 4, respectively. If all possible values have been selected then STOP, the procedure is completed. The best of computed solutions is the optimal one.

Step 2. For each set of initial values selected at Step 1, calculate simultaneously from left to right the primal equations (7), (10) and dual equations (27), (26) for the parallel and serial systems, respectively. For every machine m and product j, the respective term of the Hamiltonian is calculated at every time point t_n as follows:

$$H_{mj}(t_n) = P_{mj}\psi_j(t_n) - S(P_{mj}) \text{ (for the parallel production system)}$$

$$H_{mj}(t_n) = P_{mj}(\psi_{mj}(t_n) - \psi_{m-1j}(t_n)) - S(P_{mj}) \text{ (for the serial production system)}.$$

Then, the optimal control $p_{mj}(t_n)$ for machine m on product j at time t_n is set according to Lemma 1 so that the Hamiltonian is maximized:

$$p_{mj}(t_n) = \begin{cases} P_{mj}\,, & \text{if } j = \arg\max_j H_{mj}(t_n) \text{ and } H_{mj}(t_n) > 0\,; \\ 0\,, & \text{otherwise}\,. \end{cases}$$

Step 3. Calculate objective function (9) for the parallel system and (12) for the serial system for the obtained inventory levels.

Step 4. If the objective improves, then save the result along with the obtained terminal values for the dual variables. Return to Step 1.

If no singular regimes are found on the optimal trajectory, the obtained solution provides the exact optimal solution; otherwise at points where singularity occurs the algorithm replaces such regimes with either production or no-production regimes (Step 2). As a result, terminal values of the dual variables will not be satisfied (equal to zero) for the obtained optimal solution, that indicates that the approximation was made.

It is important to note, that conditions (29) and (31) actually show that singular regimes are avoided everywhere or almost everywhere if inventory and production costs for different products are distinct and not multiple to each other. Such conditions would be satisfied quite naturally for real manufacturing cases. At the same time, the penalties for underproduction (tardiness) of a product are often multiples of those for overproduction (earliness) in real production systems. Furthermore, the exact zero inventories (prohibited by conditions (24)) are also unrealistic to happen, because zero values in Equations (7) and (10) can only be provided by controls compensating exactly all current demands. This clearly would be possible if the controls were continuously adjustable (i.e., $0 \leqslant p_{mj}(t_n) \leqslant P_{mj}$) to changing in time demands, rather than binary $p_{mj}(t_n) \in \{0, P_{mj}\}$ as stated in our problem formulations.

To illustrate the above algorithm, we consider three-machine, four-product, eleven-time points scheduling problem in a parallel production system with general type of demands, quadratic production cost and period $\Delta = 1$:

$$\begin{aligned} \text{minimize} \sum_{t=0}^{10} \left(\sum_{j=1}^{4} C(X_j(t)) \right. \\ &+ 2.6p_{11}^2(t) + 0.1p_{12}^2(t) + 2.5p_{13}^2(t) + 2.9p_{14}^2(t) \\ &+ 2.3p_{21}^2(t) + 2.1p_{22}^2(t) + 2.7p_{23}^2(t) + 2.6p_{24}^2(t) \\ &+ 1.0p_{31}^2(t) + 2.3p_{32}^2(t) + 2.6p_{33}^2(t) + 1.7p_{34}^2(t)) \end{aligned}$$

subject to

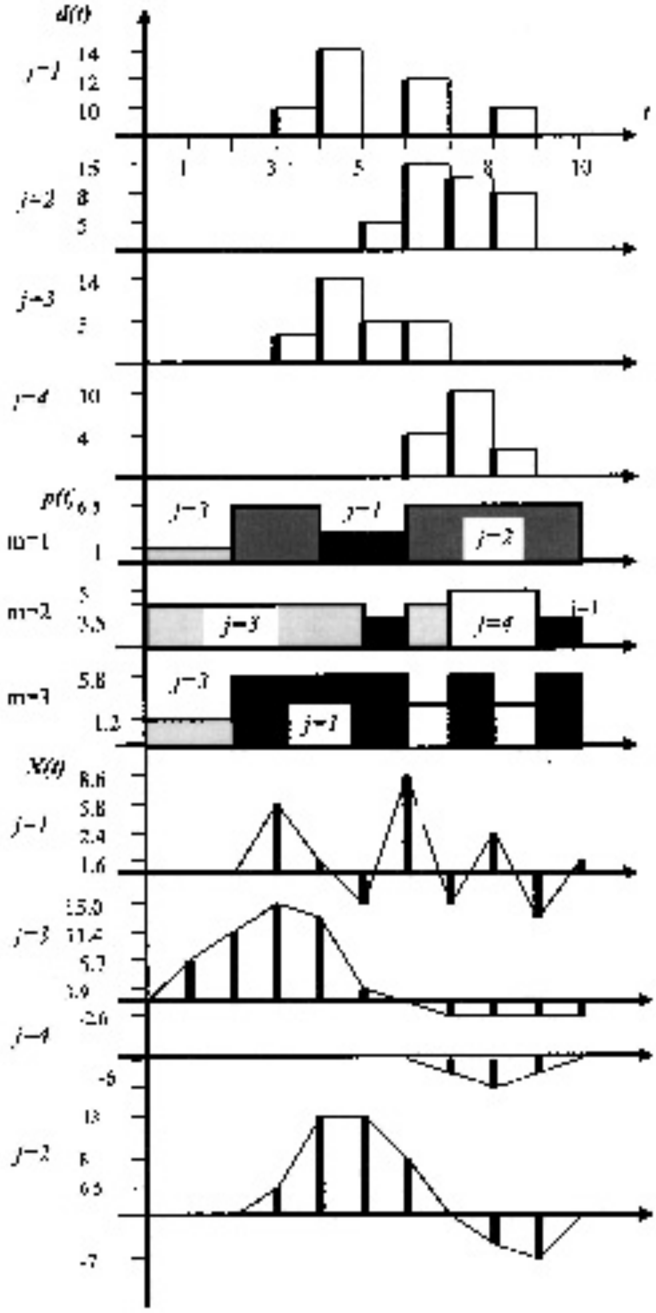

Figure 3. Production rates and inventory levels for the calculated example.

$$C(X_j(t)) = \begin{cases} |X_j(t)| \,, & \text{if } X_j(t) > 0 \\ 2|X_j(t) \,, & \text{if } X_j(t) < 0 \end{cases}, \quad j = 0, 1, \ldots, 10 \,; \; X_j(t) \neq 0 \,;$$

$$X_j(t+1) = X_j(t) + \sum_{m=1}^{3} p_{mj}(t) - d_j(t) \,, \; X_j(0) = 0.0001 \,,$$

$$t = 0, 1, \ldots, 9 \,; \; j = 1, \ldots, 4 \,;$$

$$\frac{p_{11}(t)}{3.0} + \frac{p_{12}(t)}{6.5} + \frac{p_{13}(t)}{1.0} + \frac{p_{14}(t)}{3.3} \leqslant 1 \,, \quad t = 0, 1, \ldots, 10 \,;$$

$$\frac{p_{21}(t)}{3.4} + \frac{p_{22}(t)}{3.0} + \frac{p_{23}(t)}{3.5} + \frac{p_{24}(t)}{5.0} \leqslant 1 \,, \quad t = 0, 1, \ldots, 10 \,;$$

$$\frac{p_{31}(t)}{5.8} + \frac{p_{32}(t)}{2.8} + \frac{p_{33}(t)}{1.2} + \frac{p_{34}(t)}{3.0} \leqslant 1 \,, \quad t = 0, 1, \ldots, 10 \,;$$

$p_{11}(t) = 0,\ 3.0;$ $p_{12}(t) = 0,\ 65;$ $p_{13}(t) = 0,\ 1.0;$ $p_{14}(t) = 0,\ 3.3;$ $p_{21}(t) = 0,\ 3.4;$ $p_{22}(t) = 0,\ 3.0;$

$p_{23}(t) = 0,\ 35;$ $p_{24}(t) = 0,\ 5.0;$ $p_{31}(t) = 0,\ 5.8;$ $p_{32}(t) = 0,\ 2.8;$ $p_{33}(t) = 0,\ 12;$ $p_{34}(t) = 0,\ 3.0.$

The optimal solution depicted in Figure 3 is obtained for the following initial values of the dual variables:

$$\psi_1(0) = -13, \; \psi_2(0) = -5, \; \psi_3(0) = 14, \; \psi_4(0) = -9 \,,$$

while all terminal boundary conditions are satisfied:

$$\psi_1(10) = \psi_2(10) = \psi_3(10) = \psi_4(10) = 0 \,.$$

6. Conclusions

Two dynamic scheduling models for minimizing inventory and production costs in parallel and serial production systems are studied with the aid of the maximum principle. As a result, optimal production regimes are derived, and a special form of the inventory cost is found, which allows the stated problems to be replaced with a two-point boundary-value problem. Consequently, we suggest an algorithm, which solves this problem in time much faster than naive enumeration of all machines. The algorithm can be applied to systems of reasonable sizes (e.g., dozens of machines, products and time periods) with arbitrary production cost functions. In addition, conditions for the cost relationships, inventory levels and demand profiles are derived for the purpose of improving the computational power of the algorithm and providing a number of the polynomially solvable cases.

Appendix

THE *NP*-HARDNESS OF *T3P*

It does not seem to be easy to show the *NP*-completeness of T3P by modifying the proof for 3P only slightly. Recall that the proof for 3P was done in the book Garey–Johnson (1991) by the following sequence of reductions:

$$3\text{SAT} \to 3\text{DM} \to 4\text{P} \to 3\text{P}\,,$$

where 3SAT = 3-satisfiability, 3DM = 3-dimensional matching, 4P = 4-partition, 3P = 3-partition. By checking the proof, it is easy to recognize that this sequence can be changed to:

$$3\text{SAT} \to \text{X3C} \to 4\text{P} \to 3\text{P}\,,$$

where X3C = exact cover by 3-sets, which contains 3DM as a special case.

We prove the *NP*-completeness of T3P by the sequence:

$$3\text{SAT} \to \text{XT3C} \to \text{T4P} \to \text{T3P}\,,$$

where XT3C = exact triple cover by 3-sets is defined as follows.

XT3C

Instance: A finite set X with $|X| = 3g$ and a collection C of three element subsets of X such that each $c \in C$ appears three times in C.

Question: Does C contain an exact triple cover for X, that is, a subcollection $C' \subseteq C$ such that every element of X occurs in exactly three members of C'.

REMARK. Let C'' be the collection obtained from C by picking only one $c \in C$ from the three members c in C. Then XT3C has a solution if X3C for C'' has a solution, since the solution that repeats three times the solution for X3C is a solution to XT3C. However, the converse may not be true, and necessitates an independent proof for the reduction.

In the following, we prove only the part of

$$3\text{SAT} \to \text{XT3C}\,,$$

because the rest of the reduction sequence XT3C → T4P → T3P can be done in the same manner as X3C → 4P → 3P.

LEMMA. *XT3C is NP-complete*.

Proof. We reduce 3SAT to XT3C by modifying the original argument of 3SAT → X3C (pp. 50–53 of Garey–Johnson (1991)).

3SAT
Instance: Collection $D = \{d_1, d_2, \dots, d_J\}$ of clauses on a finite set U of variables such that $|d_j| = 3$ for $1 \leqslant j \leqslant J$.

Question: Is there a truth assignment for U that satisfies all clauses in D?
For each variable $u \in U$, we introduce the following $13m$ elements in X of XT3C:

$$u_0[j],\ u_1[j],\ \bar{u}_1[j],\ u_2[j],\ \bar{u}_2[j],\ u_3[j],\ \bar{u}_3[j],\ a_1[j],\ b_1[j],\ a_2[j],\ b_2[j],\ a_3[j],\ b_3[j]\,,\ 1, 2, \dots, J\,.$$

Example for the case of $J = 2$ is presented in Figure A.1.

Then, for these elements, the following 3-sets in collection C are prepared. Note that each member $c^t_{uk}[j]$, $c^f_{uk}[j]$ and $c^o_{uk}[j]$ appears three times in C:

$$c^t_{uk}[j] = \{u_k[j], a_k[j], b_k[j]\}\,, \quad j = 1, 2, \dots, J\,,\ k = 1, 2, 3\,;$$

$$c^f_{uk}[j] = \{\bar{u}_k[j], b_k[j], a_{k+1}[j]\}\,, \quad j = 1, 2, \dots, J\,,\ k = 1, 2, 3$$

$$(k + 1 \text{ is taken to be module } 3)\,;$$

$$c^o_{uk}[j] = \{u_0[j], u_k[j], \bar{u}_k[j]\}\,, \quad j = 1, 2, \dots, J\,,\ k = 1, 2, 3\,.$$

First, consider how to cover $a_k[j]$ and $b_k[j]$ for all k and j, such that each element is covered exactly three times by the above three sets. For this, there are four patterns shown in Figure A.2.

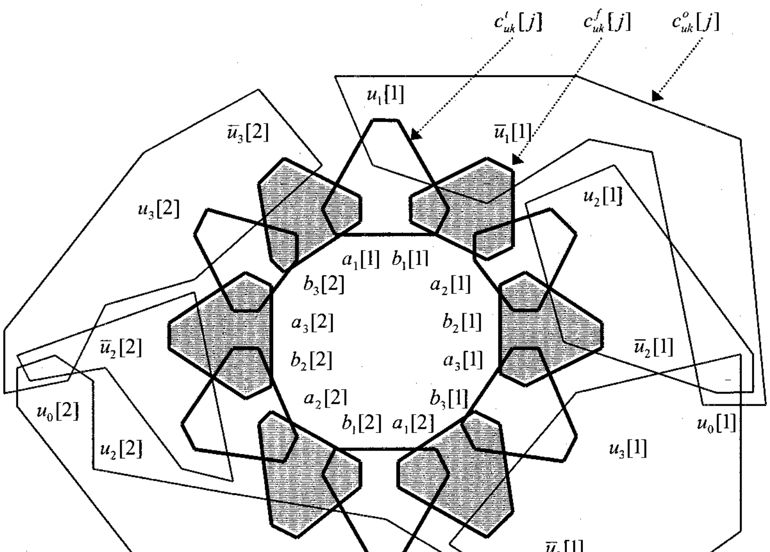

Figure A.1. Example for the case of $J = 2$.

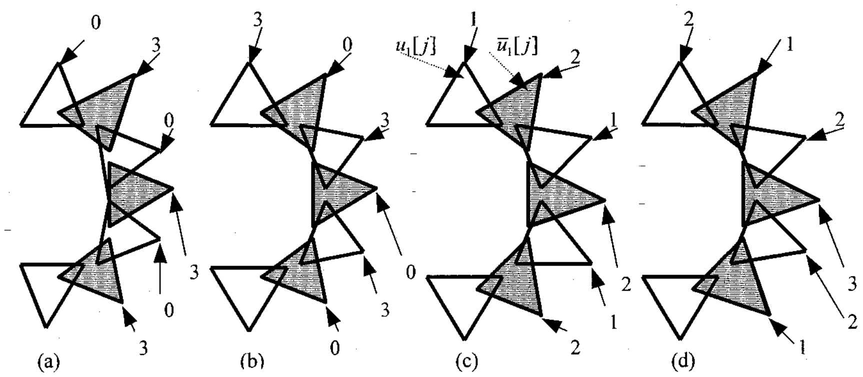

Figure A.2. Four patterns to cover $a_k[j]$ and $b_k[j]$ for all k, j.

Note, that the numbers attached to 3-sets denote how many times the sets appear in the solution.

Then we cover every $u_0[j]$ three times by choosing $c^o_{uk}[j]$ ($k = 1, 2, 3$) appropriately (i.e., three 3-sets from nine members $c^o_{u1}[j]$, $c^o_{u2}[j]$, $c^o_{u3}[j]$, where each member appears three times), while covering other elements $u_k[j]$, $\overline{u}_k[j]$, $a_k[j]$, $b_k[j]$ at most three times, respectively. It is not difficult to see that such covering is possible only if the covering pattern for $a_k[j]$ and $b_k[j]$ is either (c) or (d) (see Figure A.2), and all three members $c^o_{u1}[j]$, $c^o_{u2}[j]$, $c^o_{u3}[j]$ are chosen exactly once, respectively. We interpret that pattern (c) assigns variable u true value (i.e., $u = 1$) and pattern (d) assigns false value (i.e., $u = 0$). In case of (c), elements $u_k[j]$, $j = 1, 3, \ldots, J$, $k = 1, 2, 3$ are covered only twice, and $\overline{u}_k[j]$ are all covered three times, while in case of (d), $u_k[j]$ are all covered three times and $\overline{u}_k[j]$ are covered two times.

The rest of the construction proceeds analogously to the original proof for X3C (3DM, precisely speaking).

For each $d_j \in D$ and $k \in \{1, 2, 3\}$ we prepare the following 3-sets:

$$\{u_k[j], s_1[j], s_2[j]\}, \text{ if } u \in d_j$$
$$\{\overline{u}_k[j], s_1[j], s_2[j]\}, \text{ if } \overline{u} \in d_j .$$

Note that there are exactly three sets of this type because $|d_j| = 3$, and that each such 3-set appears three times in C, by the definition of XT3C.

Assume that the instance of 3SAT has a solution. To cover $s_1[j]$, $s_2[j]$ three times for each j, we can choose each of

$$\{u_k[j], s_1[j], s_2[j]\}, \quad k = 1, 2, 3$$

exactly once respectively, for one literal $u \in d_j$ which was assigned true value (pattern (d)). After choosing these, we note that elements $u_k[j]$ (or $\overline{u}_k[j]$), corresponding to the above choice, are covered three times.

To cover three times all the element $u_k[j]$ and $\bar{u}_k[j]$ not chosen above, we then prepare the garbage collection component. This part is almost the same as the original proof of Garey–Johnson, and is omitted.

After this construction, we can show by following the arguments of the original proof, that the instance of 3SAT has a solution if and only if the corresponding instance of XT3C has a solution.

This completes the proof for 3SAT $\rightarrow$ XT3C. □

References

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

Brucker, P. (1995), *Scheduling Algorithms*, Springer, Berlin.

Bryson, A.E. and Ho, Y.-C. (1969), *Applied Optimal Control*, Ginn and Company, Waltham.

Crowston, W.P. and Wagner, M.H. (1973), Dynamic lot-size models for multistage assembly systems, *Management Science* 20: 13–21.

Garey, M.R. and Johnson, D.S. (1991), *Computers and Intractability*, W.H. Freeman and Company, New York.

Karmarkar, U., Kekre, S. and Kekre, S. (1987), The dynamic lot-sizing problem with startup and reservation costs, *Operations Research* 35: 389–398.

Khmelnitsky, E. and Kogan, K. (1994), Necessary optimality conditions for a generalized problem of production scheduling, *Optimal Control Applications & Methods* 15: 215–222.

Khmelnitsky, E., Kogan, K. and Maimon, O. (1995), A Maximum principle based method for scheduling in a flexible manufacturing system, *Discrete Events Dynamic Systems* 5: 343–355.

Kimenia, J.G. and Gershwin, S.B. (1983), An algorithm for the computer control of a flexible manufacturing system, *IEE Transactions* 15(4): 353–362.

Kogan, K., Shtub, A. and Levit, V. (1997), DGAP – the dynamic generalized assignment problem, *Annals of Operations Research* 69: 227–229.

Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G. and Shmoys, D.B. (1993), Sequencing and scheduling: algorithms and complexity, in Graves, S., Rinnooy Kan, A. and Zipkin, P. (eds.), *Logistics of Production and Inventory*, Handbooks in Operations Research and Management Science, 4, North-Holland, New York.

Pinedo, M. (1995), *Scheduling: Theory, Algorithms, and Systems*, Prentice-Hall, New Jersey.

Sousa, J.B. and Pereira, F.L. (1992), A hierarchical framework for scheduling and planning discrete events in manufacturing systems, *Proc. Third Int. Conf. on Computer Integrated Manufacturing*, IEEE, Troy, NY, May 1992, 278–286.