Published in IIE Transactions, 2000, vol. 32, no. 10, p. 907-919
which should be cited to refer to this work

A Stochastic Programming Approach to Manufacturing Flow Control *

A. Haurie § F. Moresino

November 1, 1999

Abstract

This paper proposes and tests an approximation of the solution of a class of piecewise deterministic control problems, typically used in the modeling of manufacturing flow processes. This approximation uses a stochastic programming approach on a suitably discretized and sampled system. The method proceeds through two stages: (i) the Hamilton-Jacobi-Bellman (HJB) dynamic programming equations for the finite horizon continuous time stochastic control problem are discretized over a set of sampled times; this defines an associated discrete time stochastic control problem which, due to the finiteness of the sample path set for the Markov disturbance process, can be written as a stochastic programming problem. (ii) The very large event tree representing the sample path set is replaced with a reduced tree obtained by randomly sampling over the set of all possible paths. It is shown that the solution of the stochastic program defined on the randomly sampled tree converges toward the solution of the discrete time control problem when the sample size increases to infinity. The discrete time control problem solution converges to the solution of the flow control problem when the discretization mesh tends to 0. A comparison with a direct numerical solution of the dynamic programming equations is made for a single part manufacturing flow control model in order to illustrate the convergence properties. Applications to larger models affected by the curse of dimensionality in a standard dynamic programming techniques show the possible advantages of the method.

*This research has been supported by FNRS-Switzerland and FCAR-Quebec.
§Corresponding Author: Alain Haurie, HEC-Management Studies, Université de Genève, 102 boulevard Carl-Vogt, 1211 Genève 4, Switzerland.

1 Introduction

Piecewise deterministic control systems (PDCS) offer an interesting paradigm for the modeling of many industrial and economic processes. The theory developed by Wonham [26] or Sworder [24] for linear quadratic systems, Davis [7], Rishel [19, 20] and Vermes [25] for more general cases, has established the foundations of a dynamic programming (DP) approach for the solution of this class of problems. There are two possible types of DP equations that can be associated with a PDCS: (i) the Hamilton-Jacobi-Bellman (HJB) equations defined as a set of coupled (functional) partial differential equations (see e.g. [19, 20]); (ii) the discrete event dynamic programming equations based on a fixed-point operator à la Denardo [8] for a value function defined at jump times of the disturbance process (see e.g. [4]).

The modeling of manufacturing flow control processes has greatly benefited from the use of PDCS paradigms. Olsder & Suri [18] have first introduced this model for a flexible manufacturing cell where the deterministic system represents the evolution of parts surpluses and the random disturbances represent the machine failures and repairs. This modeling framework has been further studied and developed by many others (we cite Gershwin et al. [12], [21] and Akella & Kumar [1], Bielecki & Kumar [2] as a small sample of the large literature on these models, nicely summarized in the books of Gershwin [11] and Sethi & Zhang [22]). When the model concerns a single part system and the failure process does not depend on the part surplus and production control, an analytic solution of the HJB equations can be obtained as shown in [1]. As soon as the number of parts is two or more, an analytic solution is difficult to obtain and one has to rely on a numerical approximation technique. A solution of the HJB equations via the approximation scheme introduced by Kushner and Dupuis [16] has been proposed by Boukas & Haurie [4]. A solution of the discrete event dynamic programming equations via an approximation of the Denardo fixed-point operator has been proposed in Boukas, Haurie & Van Delft [3]. Both methods suffer from the *curse of dimensionality* and tend to become ineffective as the number of parts is three or over. Caramanis & Liberopoulos [5] have proposed an interesting approach based on the use of a sub-optimal class of controls, depending on a finite set of parameters, these parameters being optimized via an infinitesimal perturbation technique. Haurie, L'Ecuyer & Van Delft [13] have further studied and experimented such a method based on a combination of optimization and simulation.

In the present paper we propose another approach combining optimization and simulation that will be valid when the disturbance jump process does not depend on the continuous state and control. The approach exploits the formal proximity which exists, under this assumption, between the PDCS formalism and the stochastic programming paradigm introduced in the realm of mathematical programming by Dantzig & Wets [6] and further developed by many others (see the survey book of Kall and Wallace [15] or the book of Infanger [14] as representatives of a vast list of contributions). The proposed method is based on a two-step approximation: (i) the HJB dynamic programming equations for the finite horizon continuous time stochastic control problem are discretized over a set of sampled times; this defines an associated discrete time stochastic control problem which, due to the finiteness of the sample path set for the Markov disturbance process, can be written as a stochastic programming problem. (ii) The very large event tree representing the sample path set is replaced with a reduced tree obtained by randomly sampling over this sample path set. It will be shown that the solution of the stochastic program defined on the randomly sampled tree converges toward the solution of the discrete time control problem when the sample size tends to infinity. The solution of the discrete time control problem converges to the solution of the flow control problem when the discretization mesh decreases. Therefore SP methods can be implemented to solve this class of PDCS and the recent advances in the numerical solution of very large scale stochastic programs can be exploited to obtain insight for problems that fall out of reach of standard dynamic programming techniques.

The paper is organized as follows. In section 2 we recall a formulation of the manufacturing flow control problem proposed by Sharifnia [23] with the PDCS formalism and the HJB equations one has to solve in order to characterize the optimal control. In section 3 we construct the discrete time approximation leading to a stochastic programming problem which will be characterized, usually, by a very large event tree representing the uncertainties. In section 4 we show how to use a random sampling of scenarios to reduce the size of the event tree and we prove convergence of this Monte-Carlo method. In section 5 we compare different approaches on a simple single part model. In section 6 we experiment the stochastic programming approximation method on two instances of a more realistic multi-part model.

2 The manufacturing flow control problem

In this section we recall the model of a flexible manufacturing system which was proposed by Sharifnia in [23]. We have chosen this model since it was already linked to linear programming in a discrete time approximation of the solution of the manufacturing flow control problem (MFCP), in the absence of random disturbances. The random disturbances introduced in many

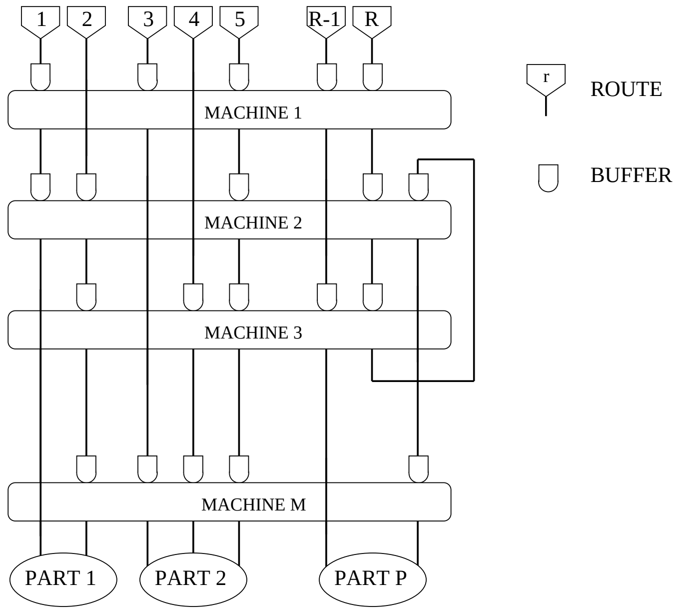


Figure 1: Flexible workshop producing P parts with M machines

formulations of the MFCP are represented as an uncontrolled Markov chain that describes the evolution of the operational state of the machines. Under these conditions, the discrete time approximation proposed in [23] will easily lend itself to a formulation as a stochastic linear programming problem.

2.1 The continuous flow formulation

We consider a flexible workshop consisting of M unreliable machines, and producing P part types. We use a continuous flow approximation to represent the production process. Each part, to be produced, has to visit some machines in a given sequence. We call this sequence a route. For a given part the route may not be unique, therefore there are R routes with $R \geq P$. An input buffer is associated with each machine. Set-up times are assumed to be negligible and processing times are supposed to be deterministic. An instance of this type of organization is represented in Figure 1. Assume that the machines are unreliable, the repair and failure times are exponentially distributed random variables. The demand is supposed to be known in advance. The objective is to minimize the expected cost associated with the work-in-process and finished parts' inventory.

For a more formal description of the model we introduce the following variables.

$$\begin{array}{rcll} v(t) & = & (v_1(t),\ldots,v_B(t))^T & : \text{ buffer processing rates} \\ w(t) & = & (w_1(t),\ldots,w_R(t))^T & : \text{ part release rates} \\ q(t) & = & (q_1(t),\ldots,q_B(t))^T & : \text{ buffer levels} \\ d(t) & = & (d_1(t),\ldots,d_P(t))^T & : \text{ finished parts demand rates} \\ y(t) & = & (y_1(t),\ldots,y_P(t))^T & : \text{ finished parts surplus levels.} \end{array}$$

The state variables are $q(t)$ and $y(t)$, while $w(t)$, $v(t)$ are the control variables. The state equations are

$$\dot{q}(t) = A_1 v(t) + A_2 w(t) \quad (1)$$

$$\dot{y}(t) = A_3 v(t) - d(t) \quad (2)$$

where the term $A_1 v(t)$ in Eq. (1) represents the internal material flows among buffers, the term $A_2(t)w(t)$ in Eq. (1) represents external arrival into the system and the term $A_3 v(t)$ represents the arrival of finished parts in the last buffer. The b-th line of A_1 is composed of a -1 in cell (b,b), and a $+1$ in cell (b,b') if the buffer b' is upstream to buffer b. All other entries of row b are 0 valued. The incidence matrix A_2 is of dimension $B \times R$, with $\{0,1\}$ entries that determine which buffers receive the new arrivals. Eq. (2) represents the dynamics of finished parts surplus. The $P \times B$ matrix A_3 has a $+1$ in entry (b,b') if buffer b' is the last buffer of a route for part b. Otherwise, this entry is 0.

Let τ_b denote the processing time of parts in buffer b and $\mathcal{B}^{(m)}$ be the set of buffers for machine m. The capacity constraints on the control are defined as follows:

$$\sum_{b \in \mathcal{B}^{(m)}} \tau_b v_b(t) \leq \xi^m(t) \quad m = 1,\ldots,M, \quad (3)$$

where $\{\xi^m(t) : t \geq 0\}$ is a continuous time Markov jump process taking the values 0 or 1. $\xi^m(t) = 1$ indicates that the machine m is operational (up) at time t, $\xi^m(t) = 0$ that it is not operational (down) at time t.

The following inequality constraints have to be satisfied

$$v(t) \geq 0 \quad (4)$$

$$w(t) \geq 0 \quad (5)$$

$$q(t) \geq 0. \quad (6)$$

Notice that (6) represents a state constraint.

Let's call

$$x(t) = (q(t), y(t))$$

the *continuous state* of the system and

$$\Xi(t) = (\xi^m(t))_{m=1,\dots,M}$$

its *operational state* while

$$u(t) = (w(t), v(t))$$

is the control at time t. The operational state $\Xi(t)$ evolves as a continuous time Markov jump process with transition probabilities that are easily computed from the failure and repair rates of each machine

$$\begin{aligned} P[\Xi(t+dt) = j|\Xi(t) = i] &= q_{ij}dt + \circ(dt) \quad (i \neq j) \\ P[\Xi(t+dt) = i|\Xi(t) = i] &= 1 + q_{ii}dt + \circ(dt) \\ \lim_{dt \to 0} \frac{o(dt)}{dt} &= 0 \end{aligned}$$

for $i, j \in I = \{0,1\}^M$. As usual we define $q_{ii} = -\sum_{i \neq j} q_{ij}$.

A production policy γ can be viewed either as

- a piecewise open-loop control $u^{\Xi(t)}(t) : t \geq 0$ that is adapted to the vector jump process $\{\Xi(t) = (\xi^m(t))_{m=1,\dots M} : t \geq 0\}$ and satisfies the constraints (3-6), when one uses a discrete event dynamic programming formalism

- a feedback law $u(t) = \gamma(t, x(t), \Xi(t))$, when one uses the coupled HJB dynamic programming equations formalism.

The variable $y^+(t) = (\max\{y_j(t), 0\})_{j=1,\dots,p}$ represents the inventory of finished parts while $y^-(t) = (\max\{-y_j(t), 0\})_{j=1,\dots,p}$ represents the backlog of finished parts. The objective is to find a policy γ^* which minimizes the expected total cost

$$E_\gamma[\int_0^T \{hq(t) + g^+y^+(t) + g^-y^-(t)\}\, dt], \tag{7}$$

where h, g^+ and g^- are cost-rate vectors for the work-in-process and finished parts inventory/backlog respectively.

2.2 The system of coupled HJB equations

To summarize, the optimal operation of the flexible workshop is a particular instance of a stochastic control problem

$$
\begin{aligned}
J^i(0,x_0) &= \min_{\gamma} E_\gamma[\int_0^T L(x(t))\,dt] && (8)\\
\text{s.t.} & && \\
\dot{x}(t) &= f(x(t),u(t)) && (9)\\
P[\Xi(t+dt) &= j|\Xi(t)=i] = q_{ij}dt + o(dt) \quad (i \neq j) && (10)\\
P[\Xi(t+dt) &= i|\Xi(t)=i] = 1 + q_{ii}dt + o(dt) \quad i,j \in I, && (11)\\
\lim_{dt\to 0}\frac{o(dt)}{dt} &= 0 && (12)\\
u(t) &\in U^{\Xi(t)} && (13)\\
& \Xi(0)=i, x(0)=x_0 && (14)\\
& && (15)
\end{aligned}
$$

where $L(x)$ and $f(x,u)$ satisfy the usual regularity assumptions for control problems.

Define the value functions

$$
J^i(t,x) = \min_{\gamma} E_\gamma[\int_t^T L(x(s))\,ds|x(t)=x \text{ and } \Xi(t)=i], \quad i \in I. \qquad (16)
$$

If these functions are differentiable in x, then the optimal policy is characterized by a system of coupled HJB equations

$$
\begin{aligned}
-\frac{\partial}{\partial t}J^i(t,x) &= \min_{u\in U^i}\{+L(x) + \frac{\partial}{\partial x}J^i(t,x)f(u) + \sum_{j\neq i} q_{ij}[J^j(t,x) - J^i(t,x)]\},\\
& i \in I \quad t \in [0,T[&& (17)\\
J^i(T,x) &= 0 \quad \forall x. && (18)
\end{aligned}
$$

When the value functions $J^i(t,x)$ is known, the optimal strategy $u^*(x,t,i)$ is obtained as the solution of a set of "static" optimization problems

$$
\min_{u\in U^i}\frac{\partial}{\partial x}J^i(t,x)f(u). \qquad (19)
$$

In the case of our MFCP these problems reduce to simple linear programs.

The value function differentiability issue can be addressed through the use of the so-called viscosity solution (see e.g. Fleming and Soner [9]). The following result is established in Ref. [17].

Theorem 1. *The optimal value function is obtained as the unique viscosity solution to the system of coupled HJB equations (17)-(18).*

3 A stochastic linear programming reformulation

In this section we define a stochastic programming problem that will be used to approximate the solution of the MFCP under study.

3.1 A discrete time reformulation

We discretize time as in [23]. This permits us to replace the continuous time state equation with a difference equation and to approximate the continuous time Markov chain by a discrete time Markov chain. Let t_k denote the k-th sampled time point $k = 0, 1, \ldots, K$ with $t_0 = 0$ and $t_K = T$, $\delta t_k = t_k - t_{k-1}$, $\tilde{q}(k) := q(t_k)$, $\tilde{y}(k) := y(t_k)$, $\tilde{w}(k) := w(t_k)$, $\tilde{v}(k) := v(t_k)$, $\tilde{\xi}^m(k) := \xi^m(t_k)$ and replace the differential state equations with the difference equations:

$$
\begin{aligned}
\tilde{q}(k) &= \tilde{q}(k-1) + \delta t_k A_1 \tilde{v}(k) + \delta t_k A_2 \tilde{w}(k) \\
\tilde{y}(k) &= \tilde{y}(k-1) + \delta t_k A_3 \tilde{v}(k) - \delta t_k \tilde{d}(k),
\end{aligned}
$$

for $k = 1, \ldots, K$. The control and state constraints become

$$
\begin{aligned}
\sum_{b \in \mathcal{B}^{(m)}} \tau_b \tilde{v}_b(k) &\leq \tilde{\xi}^m(k) \quad m = 1, \ldots, M, \\
\tilde{v}(k) &\geq 0 \\
\tilde{w}(k) &\geq 0 \\
\tilde{q}(k) &\geq 0 \quad , \quad k = 1, \ldots, K \\
\tilde{q}(0) &= \tilde{q}_0 \\
\tilde{x}(0) &= \tilde{x}_0.
\end{aligned}
$$

Denote $\tilde{x}(k) = (\tilde{q}(k), \tilde{y}(k))^T$ the continuous state variables, $\tilde{u}(k) = (\tilde{w}(k), \tilde{v}(k))^T$ the control variables and $\tilde{\Xi}(k) = (\tilde{\xi}^m(k)_{m=1\ldots M})$ the discrete state variable that evolves according to a Markov chain with transitions probabilities

$$
\begin{aligned}
P[\tilde{\Xi}(k+1) = j | \tilde{\Xi}(k) = i] &= q_{ij} \delta t_k \quad (i \neq j) \\
P[\tilde{\Xi}(k+1) = i | \tilde{\Xi}(k) = i] &= 1 + q_{ii} \delta t_k.
\end{aligned}
$$

This time discretization can be envisioned when the average times to repair and failure are much greater than δt_k. The solution of the associated discrete time stochastic control problem can be obtained through the solution of the following discrete time DP equations:

$$
\begin{aligned}
&\tilde{J}^i(k-1, \tilde{x}(k-1)) \\
&= \min_{\tilde{u}(k) \in U^i} \{L(\tilde{x}(k))\delta t_k + \sum_{j \neq i} q_{ij} \delta t_k \tilde{J}^j(k, \tilde{x}(k)) + (1 + q_{ii} \delta t_k) \tilde{J}^i(k, \tilde{x}(k))\} \quad (20)
\end{aligned}
$$

for $i \in I$ and $k = 1 \dots K$; with terminal conditions:

$$\tilde{J}^i(K, \tilde{x}(K)) = 0. \tag{21}$$

The following result can be established, using standard techniques of approximation of viscosity solutions (see [17]).

Theorem 2. *The solution of the discrete time DP equation (20,21) converges uniformly when $\delta t_k \to 0$ to the viscosity solution of the system of coupled HJB equation of the continuous time model (17,18).*

3.2 The scenario concept

Since the disturbance Markov jump process is uncontrolled, the solution of the discrete time stochastic control problem can also be obtained via the so-called *stochastic programming* technique. This is a mathematical programming technique based on the concept of a scenario. For our problem we call *scenario* ω a sample path $\{(\tilde{\xi}^1_\omega(1), \dots, \tilde{\xi}^M_\omega(1)), \dots, (\tilde{\xi}^1_\omega(K), \dots, \tilde{\xi}^M_\omega(K))\}$ of the $\tilde{\Xi}(\cdot)$ process. On a time horizon of K periods, as the state in the first period is identical for all scenarios, the discrete time Markov chain will generate $(2^M)^{K-1}$ different scenarios. We denote $u_{\omega_\ell}(k)$ the control for period k when the realized scenario is ω_ℓ.

For two scenarios ω_ℓ and $\omega_{\ell'}$ that satisfy

$$(\tilde{\xi}^1_{\omega_\ell}(k), \dots, \tilde{\xi}^M_{\omega_\ell}(k)) = (\tilde{\xi}^1_{\omega_{\ell'}}(k), \dots, \tilde{\xi}^M_{\omega_{\ell'}}(k)) \quad \forall k \leq l \tag{22}$$

the controls $u_{\omega_\ell}(k)$ and $u_{\omega_{\ell'}}(k)$ must be equal for all $k \leq l$. These conditions are called the *nonanticipativity constraints.*

There are two possible ways to take these constraints into account in the optimization problem

(i) introduce as many subproblems as there are scenarios and couple them through the nonanticipativity constraints explained above,

(ii) handle the scenario tree on a node by node basis with the nonanticipativity constraint taken into account implicitly.

The second approach is usually preferable because it reduces the number of constraints in the associated mathematical program. Let $\mathcal{N}(k) = \{\mathcal{N}_1(k), \dots, \mathcal{N}_{\ell_k}(k)\}$ be the set of the nodes at period k. For each scenario ω and for each period k, ω passes through one and only one node $\mathcal{N}_\ell(k)$ (that we denote $\omega \hookrightarrow \mathcal{N}_\ell(k)$). If ω_ℓ and $\omega_{\ell'}$ are indistinguishable until the period l, that is if (22) holds, then they share the same node $\mathcal{N}_\ell(k)$ at all periods

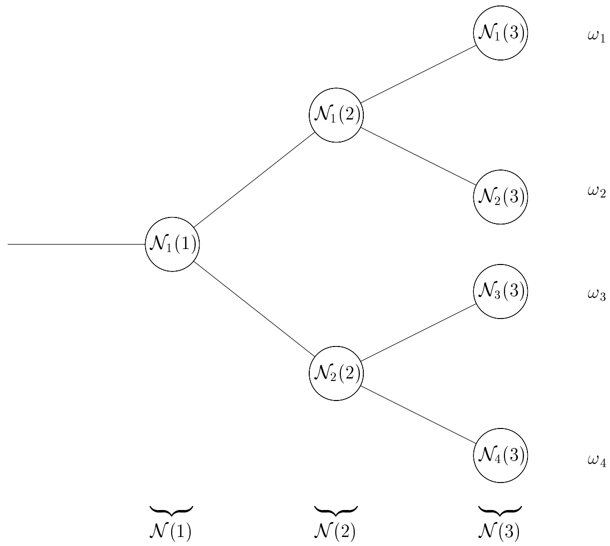


Figure 2: The scenario tree of a workshop with 1 machine and 3 periods

$k \leq l$. Note that since all scenarios are indistinguishable in the first period, we have only one node for this period, e.g. $\mathcal{N}(1) = \{\mathcal{N}_1(1)\}$. Each node n, except $\mathcal{N}_1(1)$ noted n_1, has a direct ancestor, denoted $\mathcal{A}(n)$, in the set of the nodes of the previous period. If ω passes through $\mathcal{N}_\ell(k)$ at period $k > 1$, then it passes through the ancestor of $\mathcal{N}_\ell(k)$ at period $k-1$. The set of all scenarios passing through the node $\mathcal{N}_\ell(k)$ is denoted by $N_\ell(k)$. The probability of the node $\mathcal{N}_\ell(k)$ is then

$$p(\mathcal{N}_\ell(k)) = \sum_{\omega \hookrightarrow N_\ell(k)} p(\omega)$$

where $p(\omega)$ denotes the probability of the scenario ω. We must then index each variable on the node set: $\tilde{q}_n(k)$, $\tilde{y}_n(k)$, $\tilde{i}_n(k)$, $\tilde{v}_n(k)$, $\tilde{\xi}_n^m(k)$ for all $n \in \mathcal{N}(k)$.

To illustrate this representation, consider a workshop of one machines with an horizon of 3 periods. In the first period the machine is up. There exist 4 scenarios which are listed in Figure 2. In the scenario ω_1 the machine is up during all periods. In the scenario ω_2 (resp. ω_3) the machine is up during all periods except period 3 (resp. period 2). In the last scenario ω_4, the machine is down during all periods except period 1. For example, the scenario ω_2 is defined by $(\tilde{\xi}(1), \tilde{\xi}(2), \tilde{\xi}(3)) = (1, 1, 0)$. $\mathcal{N}(2)$, the set of nodes at period 2, contains two nodes: $\mathcal{N}_1(2)$ and $\mathcal{N}_2(2)$. The direct ancestor of $\mathcal{N}_2(3)$ is $\mathcal{N}_1(2)$.

3.3 A linear stochastic programming problem

To summarize, we have to solve a stochastic linear program with the objective function

$$\tilde{J}^{0,\tilde{\Xi}}(\tilde{x}_0) = \min \sum_{k=1}^{K} \sum_{n\in\mathcal{N}(k)} p(n)\{h\tilde{q}_n(k) + g^+\tilde{y}_n^+(k) + g^-\tilde{y}_n^-(k)\}\delta t_k. \quad (23)$$

For the first period the constraints are

$$\begin{aligned} \tilde{q}_{n_1}(1) &= \tilde{q}_0 + \delta t_1 A_1 \tilde{v}_{n_1}(1) + \delta t_1 A_2 \tilde{w}_{n_1}(1) && (24)\\ \tilde{y}_{n_1}(1) &= \tilde{y}_0 + \delta t_1 A_3 \tilde{v}_{n_1}(1) - \delta t_1 \tilde{d}_{n_1}(1), && (25)\\ \sum_{b\in\mathcal{B}^{(m)}} \tau_b(\tilde{v}_{n_1})_b(1) &\le \tilde{\xi}^m_{n_1}(1) \quad m = 1,\dots,M, && (26)\end{aligned}$$

with the initial conditions

$$\begin{aligned} (\tilde{\xi}^m_{n_1}(1))_{m=1,\dots,M} &= \tilde{\Xi}\\ (\tilde{q}_0, \tilde{y}_0) &= \tilde{x}_0. \end{aligned}$$

For each period $k = 2\dots K$ the following constraints must hold for $n \in \mathcal{N}(k)$:

$$\begin{aligned} \tilde{q}_n(k) &= \tilde{q}_{\mathcal{A}(n)}(k-1) + \delta t_k A_1 \tilde{v}_n(k) + \delta t_k A_2 \tilde{w}_n(k) && (27)\\ \tilde{y}_n(k) &= \tilde{y}_{\mathcal{A}(n)}(k-1) + \delta t_k A_3 \tilde{v}_n(k) - \delta t_k \tilde{d}_n(k), && (28)\\ \sum_{b\in\mathcal{B}^{(m)}} \tau_b(\tilde{v}_n)_b(k) &\le \tilde{\xi}^m_n(k) \quad m = 1,\dots,M. && (29)\end{aligned}$$

For $k = 1\dots K$ and $n \in \mathcal{N}(t)$ the following non-negativity constraints must hold.

$$\tilde{v}_n(k), \quad \tilde{w}_n(k), \quad \tilde{q}_n(k) \ge 0. \quad (30)$$

The optimal policy $\tilde{\gamma}^*$ is then described by the controls $\tilde{u}_n(k) = (\tilde{w}_n(k), \tilde{v}_n(k))$ for $n \in \mathcal{N}(t)$.

3.4 Identification of hedging points

The stochastic programming formulation will be used primarily for a computation of the control law at the initial time 0. Using parametric analysis we will be able to identify a suboptimal policy for running the FMS in a stationary (ergodic) environment. The optimal control for an MFCP is often an hedging point policy. In the continuous time, infinite horizon HJB formulation the hedging point corresponds to the minimum of the (potential) value function. In a finite time horizon formulation, the minimum of the value function at time 0 will tend to approximate the optimal hedging point when the horizon increases. In our discrete time, finite horizon formulation, if we let the initial stocks $\tilde{q}_0$ and $\tilde{y}_0$ be free variables, their optimal values will give an indication of the hedging points. Actually, the discretization of time will often elimininate the uniqueness of the hedging points defined as the minimum of the value function. It will be then useful to identify the hedging point as the initial state for which the actual optimal value of production is exactly equal to the demand.

4 Approximating the stochastic linear program by sampling

In this section we propose a sampling technique to reduce the size of the stochastic programming problem one has to solve to approximate the control policy.

4.1 The approximation scheme

To solve the linear stochastic program introduced in section 3, we have to consider the event tree representing the $(2^M)^{K-1}$ different possible scenarios. This number of possible scenarios increases exponentially with the number of periods and the problem becomes rapidly intractable. To reduce the size of the problem we extract a smaller event tree composed of randomly sampled scenarios.

Only the control for the first period is really relevant and we want to find the optimal policy $\gamma^*(t, x(t), \Xi(t))$ for $t = 0$. We will solve the sampled stochastic programming model for different initial states $\tilde{x}_0$ on a given finite grid G. Then the control $\gamma^*(0, x(0), \Xi(0))$ is approximated by $\tilde{u}^*(1)$, the solution for the first period in the sampled stochastic programming model when $\tilde{\Xi} = \Xi(0)$ and where $\tilde{x}_0$ is the nearest point to $x(0)$ in G.

4.2 Convergence of the sampled problems solutions

Let us introduce a few simplifying notations. Consider a discrete probability space $(\Omega, \mathcal{B}, P)$, where Ω is the finite set of possible realizations ω of the uncertain parameters and P the corresponding probability distribution. As Ω is finite, the event set is $\mathcal{B} = 2^\Omega$. Let $S = |\Omega|$ be the number of different scenarios. The elements of Ω are denoted $\Omega = \{\omega_1, \ldots, \omega_S\}$. Let $p(\omega_i)$ denote the probability of the realization ω_i. A generic stochastic optimization problem can be represented as a convex optimization problem (here x and y are used to represent generic variables in an optimization problem; they don't have the signification given to them in the MFCP)

$$z = \min \sum_{\omega \in \Omega} f(x, \omega) p(\omega) \tag{31}$$

s.t.

$$x \in C \subseteq \mathbf{R}^n \tag{32}$$

We assume that $f(x, \omega)$ is convex in x on the convex set C but not necessarily differentiable. This formulation (31, 32) encompasses the classical two-period stochastic program with recourse

$$f(x, \omega) = cx + \min_y C(\omega) y$$

$$\begin{aligned} s.t \quad & D(\omega) y = d(\omega) + B(\omega) x \\ & y \geq 0 \end{aligned}$$

$$C = \{x \in \mathbf{R}^n | \quad Ax = b, \quad x \geq 0\}$$

In this formulation the variable x represents the decision in the first period and y is the recourse in second period. Once the optimization w.r.t. y has been done for each possible realization ω, the problem is reduced to the form (31, 32).

The stochastic programming problem obtained from the time discretization of the MFCP can also be put in the general form (31, 32) through a nested reduction of a sequence of two stage stochastic programming problems. The variable x will then represent the decision variables for the initial period (the one we are particularly interested in).

We now formulate an approximation of the generic problem obtained through a random sampling scheme. A sampled problem, with sample size m, is obtained, if we draw randomly m scenarios among the S possible. A specific scenario ω_i is selected at a given draw with probability $p(\omega_i)$. We denote $\varpi^m = \{\varpi_j, j = 1, \ldots, m\}$, the scenario sample thus obtained. The sampled SP problem is defined as

$$z^{\varpi^m} = \min_x \frac{1}{m} \sum_{j=1}^{m} f(x, \varpi_j) \tag{33}$$

s.t.

$$x \in C \subseteq \mathbf{R}^n. \tag{34}$$

Let ν_i be the observed frequency of scenario ω_i in the sample ϖ^m. If we denote by $w_i = \frac{\nu_i}{m}$ the observed proportion of scenario ω_i, the problem (33, 34) can also be reformulated as

$$z^{\varpi^m} = \min_x \sum_{i=1}^{S} f(x, \omega_i) w_i \tag{35}$$

s.t.

$$x \in C \subseteq \mathbf{R}^n. \tag{36}$$

The convergence of the sampled problem solution to the original solution is stated in the following theorem.

Theorem 3. *When $m \to \infty$ the solution z^{ϖ^m} of the sampled stochastic optimization problem (33, 34)converges almost surely to the solution z of the original stochastic optimization problem (31, 32).*

Proof. According to the strong law of large numbers we know that the observed proportions $(w_i)_{i=1,\ldots,S}$ converge almost surely to the probabilities $p(\omega_i)_{i=1,\ldots,S}$ when the sample size m tends to infinity. Furthermore, one can easily show that the function $\min_{x \in C} \sum_{i=1}^{S} f(x, \omega_i) p_i$ is convex, and therefore continuous, in $(p_i)_{i=1,\ldots,S} \in \mathbf{R}^S$. These two properties lead to the desired result. □

5 Empirical verification of convergence

In this section we illustrate the convergence of the SP method on a single-machine single-part-type MFCP which is the finite horizon counterpart of the example treated by Bielecki and Kumar [2]. A solution for the finite horizon case has been proposed in [27] under the rather strong assumption that once the machine fails it will never be repaired. In the general case with finite horizon there is no analytical solution available, however a direct numerical solution of the HJB equations can be obtained with good accuracy, using the weak convergence technique proposed by Kushner and Dupuis [16]. This alternative numerical solution will be used to control the convergence of our sampled SP models.

Indeed for this example the direct solution of the dynamic programming equations is more efficient than the sampled SP method. However, when there are two or more part-types we expect the sampled SP method to be more efficient than the direct dynamic programming method.

The problem is:

$$\min_{\gamma} E_{\gamma}\left[\int_0^T L(x(t))\,dt\right]$$

$$\begin{aligned}
s.t. \quad & \dot{x}(t) = u(t) - d \\
& P[\Xi(t+dt) = j\,|\Xi(t) = i] = q_{ij}dt + \circ(dt) \quad (i \neq j) \\
& P[\Xi(t+dt) = i|\Xi(t) = i] = 1 + q_{ii}dt + \circ(dt) \\
& u(t) \in U^{\Xi(t)} \quad U^0 = \{0\} \quad U^1 = [0, u_{\max}] = [0, \frac{1}{\tau}] \\
& \Xi(0) = i \in \{0, 1\} \\
& x(0) = x_0
\end{aligned}$$

With $L(x(t)) = g^+x^+(t) + g^-x^-(t)$.

An accurate numerical solution can be obtained via a direct solution of the dynamic programming equations. This numerical solution shows that for the finite-time horizon the optimal control is an hedging point policy but with a safety stock that decreases when one gets closer to the end of horizon T, i.e.

$$u^*(x,t) = \begin{cases} u_{\max} & \text{if } x < Z(T-t) \\ d & \text{if } x = Z(T-t) \\ 0 & \text{if } x > Z(T-t), \end{cases}$$

where $Z(\cdot)$ is an increasing function called the *hedging curve*.

5.1 Accuracy of the SP solution

We solve the finite horizon model with the following data ([11] p.292): $g^+ = 1$, $g^- = 10$, $d = 0.5$, $q_{01} = 0.09 = -q_{11}$, $q_{10} = 0.01 = -q_{00}$ and $\tau = 1$.

To control the convergence of our SP solution, we implemented the method of Kushner and Dupuis ([16] Chapter 12) on the x-state space grid

$$G = \{-30, -29.99, -29.98, \ldots, 70\}$$

and with a time step 0.001. For the infinite horizon case, the hedging point is $Z = 4.9279$ (see Ref. [2]). The solid line in Figure 3 is the hedging curve obtained via the Kushner and Dupuis numerical technique. One notices that, as expected, the hedging curve tends asymptotically to the hedging point value 4.9279 when the horizon increases.

The size of the associated stochastic programming model increases exponentially with the number of periods K. The largest possible value K

permitted by the memory on our machine (IBM RISC 6000, with 128 Mb memory, running SP/OSL software) was equal to 13 corresponding to 4096 different scenarios. For the computations concerning a model with more than 13 periods, we applied the following recursive method. We first compute the value functions $\tilde{J}^0(0,\cdot)$ and $\tilde{J}^1(0,\cdot)$, defined in Equation (23), for 13 periods. Then, in the objective function, a piecewise linear approximation of each value functions is introduced as a terminal cost penalty. The value functions of this new model are computed and a piecewise linear approximation of each of this new value function is introduced in the objective function. We can repeat this recursive procedure as often as desired. This corresponds to a value iteration on a two stage dynamic programming process.

In the SP approach we have identified the initial stock for which the optimal policy in the first period is to produce the same amount as the demand d. As noticed previously these values correspond to the hedging points. We notice that the time discretization yields an approximation of the exact hedging curve by a discontinuous function which remains however quite close to $Z(T-t)$.

Figure 3 compares the value of $Z(T-t)$ obtained via three different methods

- the solid line corresponds to the solution of the dynamic programming equations obtained via the Kushner and Dupuis method.
- the dashed line shows the solution obtained with the SP method where $\delta t = 3.0$.
- the dotted line shows the solution obtained with the SP method where $\delta t = 2.0$.

It can be observed that the hedging curve $Z(T-t)$ is approximated in the SP approach by a discontinuous function with values $\delta t \cdot d \cdot I$ where I is an integer and d is the demand rate.

In Figure 4 we have represented the value function $J^1(T-t,x)$ with a solid line, when evaluated by a direct solution of the DP equations and a dashed line when evaluated through the SP approach with $K=13$.

5.2 Accuracy of the SP solution with sampling

We investigate now the convergence of the solution of the SP method with sampling to the solution of the SP method when the whole scenario tree is taken into account. For all numerical experiments of this subsection we use $\delta t = 3$. As we noticed in the previous subsection, the approximation of the

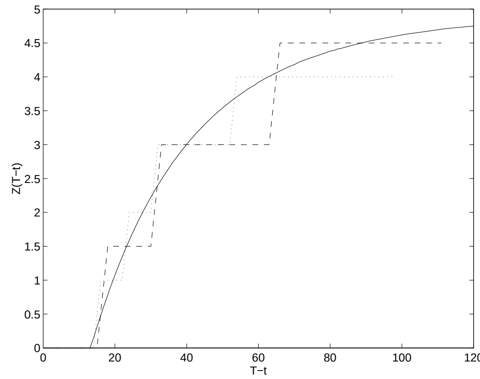
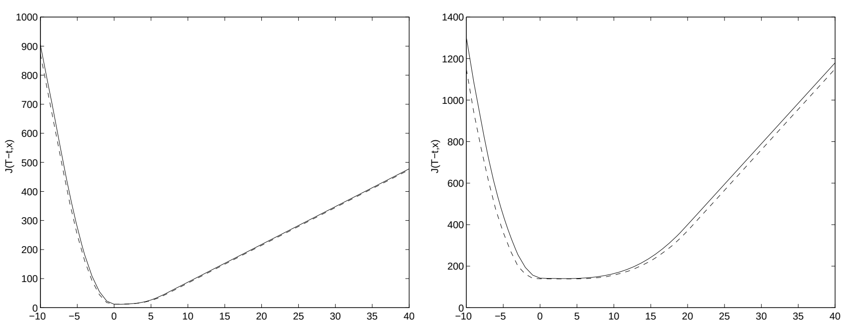


Figure 3: The hedging curve $Z(T-t)$



(a) $T-t=13$

(b) $T-t=39$

Figure 4: $J^1(T-t,x)$ versus x. SP dashed.

	sample size: 500	sample size: 5000	sample size: 50000
sample no.1	3	3	3
sample no.2	0	3	3
sample no.3	0	1.5	3
sample no.4	1.5	3	3
sample no.5	3	3	3
sample no.6	1.5	3	3

Figure 5: $Z(T-t)$ for $T-t=33$.

	sample size: 500	sample size: 5000
sample no.1	3	3
sample no.2	3	3
sample no.3	3	3
sample no.4	6	3
sample no.5	3	3
sample no.6	1.5	3

Figure 6: $Z(T-t)$ for $T-t=39$

hedging curve $Z(T-t)$ obtained with the SP reformulation is a step function. Consequently there are times to go $T-t$ at which $Z(T-t)$ is discontinuous. Therefore we investigated the SP method with sampling for the computation of $Z(T-t)$ at two possible values of the time to go $T-t$: one near a discontinuity ($T-t=33$) and one far from a discontinuity ($T-t=39$). We took different sample sizes to construct the approximating event tree and the results are shown in Figure 5 for the time to go $T-t=33$ and in Figure 6 for the time to go $T-t=39$. We see that a sample size of 500 is not sufficient for both cases. A sample size of 5000 is sufficient for $T-t=39$ but not for $T-t=33$. However a sample size equal to 50000 is sufficient for $T-t=33$.

6 Numerical experiments

In this section we apply the numerical method presented in this paper to two examples that are closer to a real life implementation. In the first subsection we approximate the optimal strategy for a flexible workshop with two machines and two part types. As the size of the model is not too big, we display the optimal strategy in full details and discuss the results. In the second subsection we study a larger system, namely a flexible workshop with six machines and four part types. Due to the size of the model, the optimal strategy cannot be fully displayed in a simple figure and therefore only the optimal hedging stocks are given.

6.1 Implementation

Our approximation scheme leads to the solution of a stochastic program. To generate and solve the stochastic program we coupled two software tools: AMPL and SP/OSL. AMPL [10] is a modeling language for mathematical programming, which is designed to help formulate models, communicate with a variety of solvers, and examine the solutions. SP/OSL is an interface library of C-language subroutines that supports the modeling, construction and solution of stochastic programs.

We obtain the solution of the stochastic program in four steps.

(i) We describe the flexible workshop topology using the algebraic facilities of AMPL. First we model the flexible workshop without the stochasticity on the machine availability (all machines are always up). This corresponds to a single scenario from the scenario tree which is from now on called the *base case scenario*.

(ii) The base case scenario is passed, in an MPS file, to SP/OSL and the whole stochastic program is constructed by specifying for every possible scenario the difference with the base case scenario and its probability or its sampled frequency. All scenarios with null probability are discarded. The sampled scenarios are then aggregated into a scenario tree.

(iii) The stochastic program is solved with SP/OSL routines, which implement a Benders decomposition.

(iv) The results are graphically displayed using MATLAB.

6.2 Two-machine two-part-type example

The example considered is a flexible workshop composed of two machines producing two parts. One operation has to be performed on each part on either the first machine or the second one, thus there are four routes. The first machine is specialized on the first part and the second machine is specialized

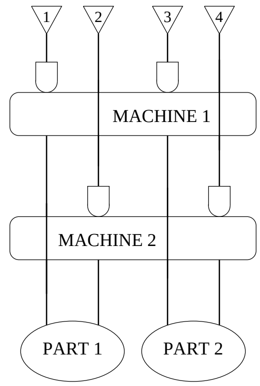


Figure 7: Flexible workshop producing two parts with two machines

on the second part. The processing time τ for each part, is equal to 0.004 for the specialized machine and 0.008 for the other machine. The penalty for work-in-process, for finished part inventory and backlog are the following:

$$\begin{aligned} h &= (1,1,1,1) \\ g^{+} &= (5,5) \\ g^{-} &= (8,8). \end{aligned}$$

The failure rate is equal to 0.3 for the first machine and 0.1 for the second one. The repair rate is equal to 0.8 for the first machine and 0.5 for the second one. The demand is supposed to be constant at 200 units per period for each part. The flexible workshop is represented in Figure 7. We consider a time horizon $T = 8$ with $K = 8$ periods. The total number of possible scenarios is about 16000 and we took as sample size $m = 10000$.

For this simple example, as only one operation has to be performed on each part, it is penalizing to have non-zero inventory in the internal buffers. So the state $\tilde{x}(k)$ is reduced to $\tilde{y}(k)$ and the policy $\tilde{u}(k)$ is fully determined by $\tilde{v}(k)$. For the finite grid G approximating $\tilde{x}_0$ we took the following values:

$$\tilde{y}_0 \in G = \{(\tilde{y}_1(0), \tilde{y}_2(0)) | \quad \tilde{y}_1(0), \tilde{y}_2(0) \in \{-200, -100, 0, 100, \ldots, 700\}\}.$$

The value function $\tilde{J}^{\tilde{\Xi}}(0, \tilde{y}_0)$ is shown in Figure 8 for $\tilde{\Xi} = (1,1)$. In this figure, we see that the value function attains a minimum on a plateau. The values of $\tilde{y}_0$ that minimize this function can be regarded as hedging points. Due to the time discretization, the set of hedging points is not, as in the continuous time case, a curve or a point, but a surface. For other values of Ξ, the value function presents the same general shape.

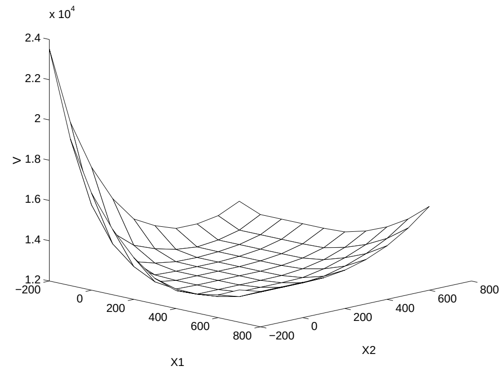
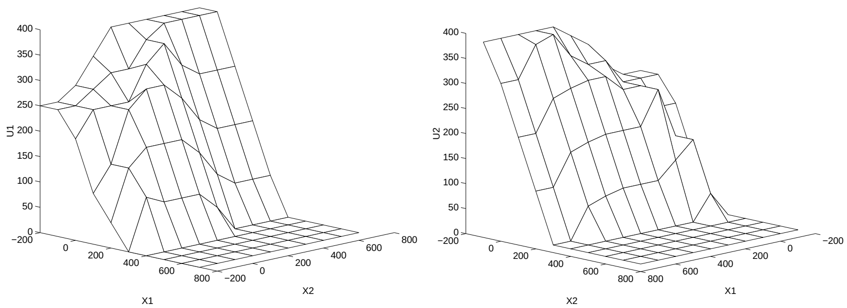


Figure 8: The function $z^*(\tilde{y}_0, \tilde{\Xi})$. In the initial period the two machines are up

For convenience, the optimal policy in the first period is rearranged as follows: the total amount of part 1 produced during the first period is denoted by $U1(\tilde{y}_0, \tilde{\Xi})$, and the total amount of part 2 produced during the first period is denoted by $U2(\tilde{y}_0, \tilde{\Xi})$. The functions $U1(\tilde{y}_0, \tilde{\Xi})$ and $U2(\tilde{y}_0, \tilde{\Xi})$ are shown in Figure 9 for $\tilde{\Xi} = (1, 1)$ and in Figure 10 for $\tilde{\Xi} = (1, 0)$.

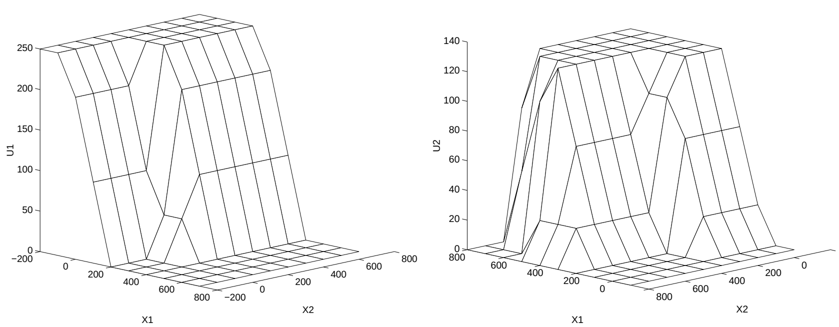
Here again we see a difference between the optimal policy of our discrete-time approximation and a typical "bang-bang" optimal policy of the continuous time model. It can be explained as follows. Suppose that for the continuous time model the optimal "bang-bang" policy is to produce at minimum rate from $t = 0$ to $t = t^*$ and then produce at maximum rate (Figure 11 top). Suppose that we discretize the time scale the same way as in section 3 with $t_{k-1} < t^* < t_k$. This optimal policy will translate on the discrete time scale as follows: produce at minimum for the periods 1 to $k - 1$, produce at maximum for the periods $k + 1$ to K and produce between minimum and maximum for the period k (Figure 11 bottom). This is clearly not a "bang-bang" policy.

An interesting result is displayed in Figure 12 which gives a cross-section of the surface shown in Figure 9 for $\tilde{y}_1(0) = 0$. We see that the priority is given to the part with the highest backlog. We see also that a high surplus of part 2 (above 300) hedges also for part 1. However this cross-hedging reaches a saturation point: a surplus of part 2 higher than 500 has the same effect as a surplus of 500.



(a) $U1(\tilde{y}_0, \tilde{\Xi})$. (b) $U2(\tilde{y}_0, \tilde{\Xi})$.

Figure 9: Optimal policy for $\Xi = (1, 1)$.



(a) $U1(\tilde{y}_0, \tilde{\Xi})$. (b) $U2(\tilde{y}_0, \tilde{\Xi})$.

Figure 10: Optimal policy for $\Xi = (1, 0)$.

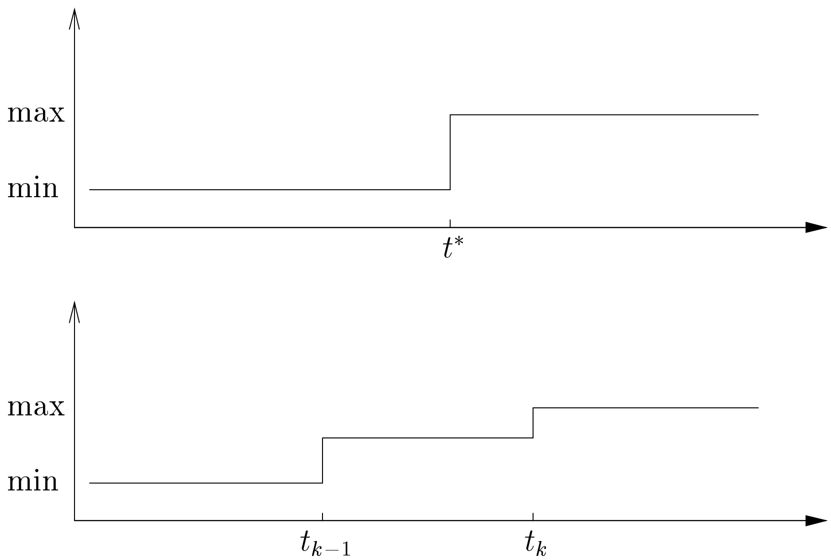


Figure 11: Effects of a time discretization on a "bang-bang" policy

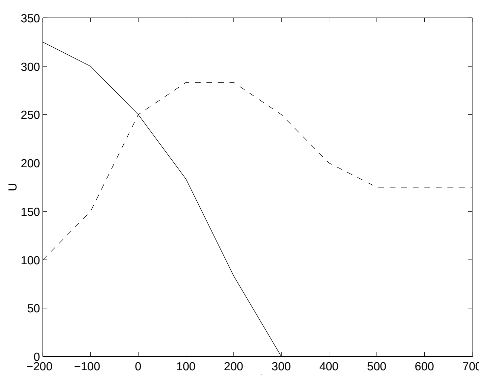


Figure 12: The functions $U1(\tilde{y}_2(0), \tilde{\Xi})$ [dotted line] and $U2(\tilde{y}_2(0), \tilde{\Xi})$ [solid line] for $\tilde{y}_1(0) = 0$. In the initial period the two machines are up

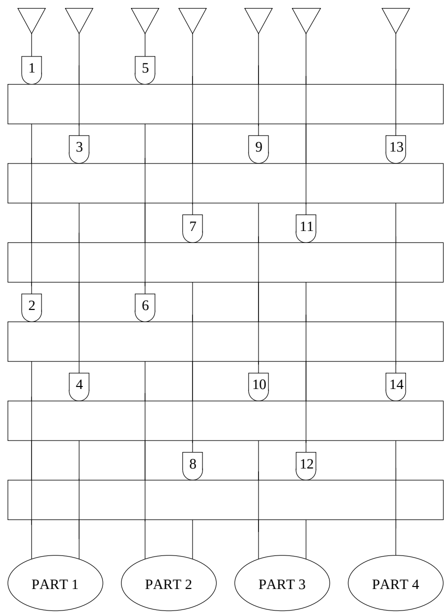


Figure 13:

6.3 Six-machine four-part-type example

The larger example considered here is a flexible workshop composed of six machines, among which 3 are unreliable, and producing four parts. The workshop topology is pictured in Figure 13. The processing time vector is given by

$$\begin{aligned}\tau \;=\; & (0.005, 0.005, 0.01, 0.015, 0.006, 0.006, 0.006,\\ & 0.006, 0.01, 0.01, 0.005, 0.005, 0.006, 0.006).\end{aligned}$$

For machines 1 and 3, the failure rate is equal to 0.1 and the repair rate is equal to 0.4. The failure rate for Machine 2 is equal to 0.2 and the repair rate is equal to 0.7. The other machines are reliable. The penalty for work-in-process equals 1 in each internal buffer; the penalty for finished part inventory (resp. backlog) equals 5 (resp. 50) for each part type. We considered a time horizon $T = 5$. The demand is supposed to be constant at 100 units for each part type.

We solved the model with $K = 5$ periods and a sample of 10000 scenarios. Given the size of the state space, it is impossible to describe the optimal policy with a simple picture. However we give in Figure 14 the hedging stocks when the six machines are operational. Since upstream from each route there is a fictive infinite buffer, we obtain, as expected, a zero hedging

Internal buffer	Hedging stock	Internal buffer	Hedging stock
buffer 1	0	buffer 2	93
buffer 3	0	buffer 4	0
buffer 5	0	buffer 6	67
buffer 7	0	buffer 8	67
buffer 9	0	buffer 10	0
buffer 11	0	buffer 12	40
buffer 13	0	buffer 14	233

Finished part buffer	Hedging stock
Part 1	100
Part 2	100
Part 3	100
Part 4	0

Figure 14:

stock for the first buffer on each route. Although we do not show the complete optimal strategy for this model, we must emphasize that it is possible to do so.

7 Conclusion

We have shown in this paper that a stochastic programming approach could be used to approximate the solution of the associated stochastic control problem in relatively large scale MFCPs. As this approach combines simulation and optimization, it can be considered as another possible method for gaining some insight on the shape of the optimal value functions that will ultimately define the optimal control. In fact, the strength of the proposed numerical method is that it is simulation based although no assumption on the nature of the optimal policy are made. Consequently the numerical approximation of the optimal strategy gives insight on the true nature of the optimal strategy. The stochastic programming approach exploits the fact that the disturbance Markov jump process is uncontrolled. It also allows the use of advanced mathematical programming techniques like decomposition and parallel processing.

References

[1] R. Akella and P.R. Kumar. Optimal control of production rate in a failure prone manufacturing system. *IEEE Transactions on Automatic Control*, 31(2):1421–1425, 1986.

[2] T. Bielecki and P.R. Kumar. Optimality of zero inventory policies for unreliable manufacturing systems. *Operations Research*, 36(4):523–541, 1988.

[3] E.A. Boukas, A. Haurie, and Ch. van Delft. A turnpike improvement algorithm for piecewise deterministic control. *Optimal Control Applications and Methods*, 12:1–18, 1991.

[4] E.K. Boukas and A. Haurie. Manufacturing flow control and preventive maintenance : A stochastic control approach. *IEEE Transactions on Automatic Control*, 35(9):1024–1031, 1990.

[5] M. Caramanis and G. Liberopoulos. Perturbation analysis for the design of flexible manufacturing system flow controllers. *Operations Research*, 40:1107–1125, 1992.

[6] G.B. Dantzig and P. Wolfe. The decomposition algorithm for linear *Econometrica*, 29:767–778, 1961.

[7] M.H.A. Davis. Piecewise deterministic markov processes: A general class of non-diffusion stochastic models. *J. of R. Stat. Soc.*, 46:353–388, 1984.

[8] E. V. Denardo. Contractions mappings in the theory underlying dynamic programming. *SIAM Review*, 9:165–177, 1967.

[9] W.H. Fleming and H.M. Soner. *Controlled Markov processes and Viscosity Solutions*. Springer Verlag, New York, Berlin, Heidelberg, 1993.

[10] R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A modeling language for mathematical programming*. The Scientific Press, San Francisco, 1993.

[11] S.B. Gershwin. *Manufacturing Systems Engineering*. PTR Prentice Hall, Englewood Cliffs, NJ, 1994.

[12] S.B. Gershwin, R. Akella, and Y. Choong. Short term scheduling of an automated manufacturing facility. *IBM Journal of Research and Development*, 29:393–400, 1985.

[13] A. Haurie, A. Leizarowitz, and Ch. van Delft. Boundedly optimal control of piecewise deterministic systems. *European Journal of Operational Research*, 73(2):237–251, 1994.

[14] G. Infanger. *Planning under Uncertainty.* The Scientific Press Series, Danvers, 1994.

[15] P. Kall and S.W. Wallace. *Stochastic Programming.* Wiley - Interscience, New York, 1995.

[16] H.J. Kushner and P.G. Dupuis. *Numerical Methods for Stochastic Control Problems in Continuous Time.* Springer-Verlag, Berlin-NY, 1992.

[17] F. Moresino, O. Pourtallier, and M. Tidball. Using viscosity solution for an approximation in piecewise deterministic control systems. Technical Report RR 3687, INRIA, Sophia-Antipolis, 1999.

[18] G.J. Olsder and R. Suri. Time optimal control of parts-routing in a manufacturing system with failure prone machines. In *Proc. IEEE International Conference on Decision and Control*, pages 722–727. 1980.

[19] R. Rishel. Dynamic programming and minimum principles for systems with jump markov disturbances. *SIAM Journal on Control*, 13(2):338–371, 1975.

[20] R. Rishel. Controlled wear process: Modeling optimal control. *IEEE Transactions on Automatic Control*, 36(9):1100–1102, 1988.

[21] Gershwin S.B., R. Akella, and Y. Choong. Performance of hierarchical production scheduling policy. *IEEE Transaction on Components, Hybrids, and Manufacturing Technology*, 7(3):225–248, 1984.

[22] S. Sethi and Q. Zhang. *Hierarchical Decision Making in Stochastic Manufacturing Systems.* Birkhäuser, Boston, 1995.

[23] A. Sharifnia. Stability and performance of distributed production control methods based on continuous-flow models. *IEEE Transactions on Automatic Control*, 39:725–737, april 1994.

[24] D.D. Sworder. Feedback control of a class of linear systems with jump parameters. *IEEE, Trans. Aut. Control*, AC-14(1):9–14, 1969.

[25] D. Vermes. Optimal control of piecewise deterministic markov process. *Stochastics*, 14:165–208, 1985.

[26] W.M. Wonham. Random differential equations in control theory. In A.T. Barucha-Reid, editor, *Probabilistic Methods in Applied Mathematics*, pages 131–217. Academic Press, NY, 1970.

[27] Q. Zhang and G. Yin. Turnpike sets in stochastic manufacturing systems with finite time horizon. *Stochastics and Stochastic Reports*, 51:11–40, 1994.