Discrete Event Dyn Syst
DOI 10.1007/s10626-006-0002-z

Optimal Threshold Levels in Stochastic Fluid Models via Simulation-based Optimization

Gül Gürkan · Fikri Karaesmen · Özge Özdemir

Received: 19 October 2004 / Accepted: 23 August 2006

Abstract A number of important problems in production and inventory control involve optimization of multiple threshold levels or hedging points. We address the problem of finding such levels in a stochastic system whose dynamics can be modelled using generalized semi-Markov processes (GSMP). The GSMP framework enables us to compute several performance measures and their sensitivities from a single simulation run for a general system with several states and fairly general state transitions. We then use a simulation-based optimization method, sample-path optimization, for finding optimal hedging points. We report numerical results for systems with more than twenty hedging points and service-level type probabilistic constraints. In these numerical studies, our method performed quite well on problems which are considered very difficult by current standards. Some applications falling into this framework include designing manufacturing flow controllers, using capacity options and subcontracting strategies, and coordinating production and marketing activities under demand uncertainty.

Keywords Stochastic optimization · Hedging points · Threshold levels · Generalized semi-Markov processes · Infinitesimal perturbation analysis · Sample-path optimization · Service-level constraints · Stochastic fluid models

The research of G. Gürkan and Ö. Özdemir reported here was sponsored by the Netherlands Organization for Scientific Research (NWO), grant 016.005.005.

G. Gürkan (✉) · Ö. Özdemir
Department of Econometrics and Operations Research, Tilburg University,
P.O. Box 90153, 5000 LE Tilburg, The Netherlands
e-mail: ggurkan@uvt.nl

Ö. Özdemir
e-mail: o.ozdemir@uvt.nl

F. Karaesmen
Department of Industrial Engineering, Koç University,
Rumeli Feneri Yolu, 34450 Sariyer, Istanbul, Turkey
e-mail: fkaraesmen@ku.edu.tr

1 Introduction

In an influential paper, Kimemia and Gershwin (1983), proposed a framework for studying production control problems as optimal control problems for stochastic fluid systems. This framework, usually referred to as manufacturing flow control, assumes that the production/inventory processes of interest are governed by flows that correspond to deterministic production and demand rates and that the randomness in the system can be captured by changes in production or demand rates according to some stochastic process governing the possible input and output rates. In recent years, there has been a strong interest in the development of models and in their analysis within this framework; Sethi et al. (2002) provide a comprehensive review of this literature.

The particular manufacturing flow control problem that we consider in this paper is that of a single resource (a "machine" for simplicity) coping with a flow of demand whose rate varies randomly. The production capacity of the machine can take multiple values according to some external stochastic process. Similarly the demand rate at any given time is governed by another external and uncontrollable stochastic process. The objective then is to dynamically select the production rate of the machine (within the capacity constraints) in order to minimize expected inventory and backorder costs. Under the general assumptions we consider here (multiple machine and demand states and general transition times), the exact solution of the optimal control problem is unknown and even if it were known the exact optimal policy is likely to be extremely complicated. On the other hand, there is a plausible class of policies called "hedging point" policies that are described by a few parameters and therefore are extremely useful for implementation in practice. A hedging point policy, in general, requires determining a parameter corresponding to each machine and demand state. The corresponding parameter (hedging point) acts as an inventory target threshold in each machine/demand state. The machine should produce at the maximum rate if the inventory level is below the current threshold, should stop if the inventory level is above the current threshold, and should produce at the rate that would keep the inventory level at the current threshold level when it reaches there.

As described further in Section 2, hedging point policies are attractive not only due to their simplicity but also because they turn out to be optimal in a number of important special cases studied in the literature. In addition, if the production policy is required to depend only on the inventory level (and not on elapsed times since previous transitions), hedging point policies seem to be the only plausible alternative. In turn, these type of policies are recently used not only for designing manufacturing controllers, but also for using capacity options and subcontracting strategies, and for coordinating production and marketing decisions.

Focusing on the class of hedging point policies reduces the optimal control problem to the optimization of a finite number of parameters: the hedging points. This optimization problem is the main focus of this paper. In particular, we propose a simulation-based approach, namely sample-path optimization, to tackle this problem. To this end, we first model the evolution of the system using a generalized semi-Markov Process (GSMP) framework. Although in the past the GSMPs were mainly used for modelling systems with discrete entities, they can also be well suited for modelling fluid systems; see Suri and Fu (1994) and Gürkan (2000) for example. Here, we also adapt the GSMP framework to our continuous-time continuous-state space problem. Utilizing the GSMP representation, we then obtain gradient estimates of the performance measures using infinitesimal perturbation analysis (IPA). Finally, we implement the relatively recent technique of sample-path optimization to find the optimal hedging points.

Sample-path optimization is a simulation-based optimization method to optimize performance functions of complex stochastic systems; it can be used for providing solutions to difficult stochastic optimization problems (including ones with stochastic constraints), stochastic variational inequalities, and equilibrium models. The basic idea is to observe a fixed sample path (by using the method of common random numbers from the simulation literature), solve the resulting deterministic problem using fast and effective methods from nonlinear programming, and then use the resulting solutions to infer information about the solution of the original stochastic problem. In Section 2, we briefly explain the underlying basic ideas and give references in which more details can be found.

Up to date, there have been several successful implementations of simulation-based optimization in the context of manufacturing flow control. Most of those papers, mentioned in Section 2, use IPA coupled with some form of stochastic approximation (SA) technique on the optimization side. More precisely, they apply SA-IPA framework to particular models and/or establish the validity of that approach for those particular models. There seems to be a number of valid reasons for focusing on particular models. To start with, proving the validity and unbiasedness of the IPA estimate can be extremely challenging and is usually dependent on the particular model representation. Moreover, the practical implementation of stochastic approximation (or its variants) has to be carefully adapted and fine tuned to the context even when unbiased derivative estimates are available. Partly as a consequence of these, the existing literature doesn't go beyond the unconstrained optimization of two hedging points.

In contrast with the most of the existing literature, we start by a fairly general single-stage model allowing any number of machine and demand states and general state transitions. Our main contributions are two-fold. First, using the GSMP framework, we obtain a flexible and general representation of the system evolution, and derive gradient estimates. Even though these estimates may be obtained by other means, the GSMP approach significantly facilitates obtaining transparent estimates by simulation and proving the unbiasedness of these estimates. Second, using sample-path optimization we avoid certain drawbacks associated with stochastic approximation type methods such as number of parameters to fine tune, slow convergence, and ad hoc handling of constraints. These two complementary features enable us to investigate much larger systems than is usually reported in comparable simulation-based optimization studies and to obtain insights on the structure of optimal hedging points for large systems. Furthermore, we also report numerical experiments with service-level type probabilistic constraints.

We would like to emphasize that although the main focus of this paper is the analysis of a particular manufacturing flow control model, the scope of application for the developed techniques is potentially much wider. Threshold based control of stochastic fluid models have useful applications in other related problems in manufacturing control (see Tan and Gershwin (2004) or Hu et al. (2004)), but also on queueing-based admission control and routing models commonly used in telecommunications (see Sun et al. 2004). The general approach proposed here may be utilized with appropriate modifications for such models as well.

The rest of the paper is organized as follows. In Section 2 we present a review of the related literature. Section 3 provides a detailed description of the problem. The GSMP framework and gradient estimation is presented in Section 4. The numerical results can be found in Section 5 and the conclusions are presented in Section 6. To improve readability, we deal with the details of our technical analysis in several appendices. Appendix 1 provides a proof of a sufficient condition for the stability of the system under consideration. Appendix 2 deals with so-called "similarity properties" needed for rigorous analysis of our gradient estimation. Appendix 3 contains the proofs of several technical results of Section 4. Finally, Appendix 4 is the pseudo-code for simulation and gradient estimation.

2 Literature

The research in manufacturing flow control can roughly be classified in two directions: papers directly addressing optimal control issues and papers that analyze and optimize the performance of plausible policies. Our work falls in the latter category but the policies we employ are strongly motivated by the research on optimal control.

Kimemia and Gershwin (1983) proposed formal optimal control theory as an approach for studying general manufacturing flow control problems. They also proposed a plausible class of control policies. The first formal proofs of optimality of a certain policy, however, was established later by Akella and Kumar (1986) and Bielecki and Kumar (1988) for a two machine state system with exponential up and down times, constant demand, linear holding and backorder costs. In particular, these papers established the optimality of a hedging point policy: whenever up, the machine should produce at full capacity below a target inventory level called the hedging point and produce at a rate which enables it to stay there once it reaches this target.

There are several extensions of the hedging point type optimality structure. For instance, for a multiple machine state system with a Markovian transition structure, a multiple hedging point type policy is optimal: for each machine state where capacity exceeds demand rate, there is a corresponding hedging point; see Sethi et al. (2002) for related results. It is also known that with non-exponential machine state transitions, the structure of the optimal policy becomes more complicated (see Hu and Xiang (1995) for an example) but hedging point policies remain useful due to their simplicity.

In addition, there is also considerable literature on the performance analysis side. Sharifnia (1988) investigates the multiple machine state problem with Markovian transitions when a multiple hedging point policy is used and presents analytical results for the performance measures of the system. Finding the optimal hedging points, however, remains difficult. Liberopoulos and Hu (1995) also investigate the multiple machine state problem with Markovian transitions with a different focus. They establish useful monotonicity properties of the optimal hedging levels with respect to the machine state transition structure. We use these intuitive monotonicity properties for testing our procedure as part of our numerical experiments in Section 5.

There are also several other examples where hedging point type policies arise in single-part and single-stage systems under more general modelling assumptions. A typical example is that of threshold subcontracting: additional manufacturing resources can be used depending on the inventory level. Tan (2002a), Hu et al. (2004), and Tan and Gershwin (2004) propose models and performance analysis approaches for this class of problems. Zhang et al. (2001) is another recent example in which joint production and marketing decisions are modelled through a hedging point type policy. Our approach, as we explain in more detail below, can be utilized in such settings with some straightforward modifications.

If one is concerned with a system with several machine and demand states and fairly general random disturbances (i.e. not necessarily exponentially distributed), then the generality of the problem and the intractability of an analytical solution makes a simulation-based method an attractive approach. Using a simulation-based method for finding optimal hedging points has been tried before by Caramanis and Liberopoulos (1992), Liberopoulos and Caramanis (1994), Haurie et al. (1994), Yan et al. (1994), Brémaud et al. (1997), Yan et al. (1999), and Yin et al. (2001). However, optimization side of all these papers were confined to the method of stochastic approximation and its variants, see Robbins and Monro (1951). Although very prominent, in practise these methods suffer from serious drawbacks such as slow convergence, lack of a good stopping criterion, and difficulty in enforcing feasibility. In particular, when there are constraints, these methods handle inequality constraints—even deterministic linear inequalities—via projection onto the feasible set. Such a projection operation can retard the performance of an optimization algorithm immensely, as illustrated by the simple example in Appendix 6 of Plambeck et al. (1996). In that example, such a method requires nearly 10^{43} steps to find the minimizer (the origin) of a linear function on the nonnegative orthant $\mathbf{R}^2_+$. Furthermore, even in the unconstrained case, the empirical performance of stochastic approximation type methods is very sensitive to the choice of a predetermined step size. Fu and Healy (1992), L'Ecuyer et al. (1994), Glasserman and Tayur (1995), and Gürkan (2000) contain a number of examples which demonstrate this sensitivity.

These difficulties are partly reflected in the available numerical results of simulation-based optimization literature for finding optimal hedging points: none of the papers we could reach (for single-part single-stage models) had numerical experiments with systems having more than two decision variables and they were all solving unconstrained problems. Furthermore, very often, following how the gradient estimates are derived is quite challenging for a non-specialist in gradient estimation literature since the arguments leading to a particular estimator are often done case by case. For example, Brémaud et al. (1997) is one of the most recent papers developing a rigorous infinitesimal perturbation analysis (IPA) algorithm for a system whose transition times are not necessarily Markovian and the IPA analysis of the two hedging point case is done in a separate section than the IPA estimate for a single hedging point.

Aside from the literature mentioned, after we have submitted this paper for publication, we have become aware of Zhao and Melamed (2004). That paper also deals with a general Make-to-Stock system (not necessarily Markovian) with random demand and production rates; IPA gradients of two performance measures are derived recursively via a sample-path analysis. However, their results are somewhat specialized (for specific initial conditions) and there are no numerical experiments reported.

To overcome the difficulties encountered in the literature, in this paper we propose using a different simulation-based method, namely sample-path optimization, coupled with a generalized semi-Markov process (GSMP) representation for the system. As we will explain below, this has two consequences. On the optimization side, we are able to solve much larger (with several machine and demand states) and general (not necessarily exponential disturbances) problems with even probabilistic constraints. On the modelling and gradient estimation side, we derive a rigorous, intuitive, and transparent algorithm for a general system that can compute the function and gradient values of the objective function and service level type probabilistic constraints at any parameter setting in a single simulation run.

Next, we describe the basic ideas behind sample-path optimization. Roughly speaking, we are concerned with solving a problem of optimization involving a limit function f_∞ which we cannot observe. However, we can use simulation to observe functions f_n that converge pointwise to f_∞ as $n \to \infty$ almost surely. In the kind of applications typically encountered, f_∞ could be a steady-state performance measure of a dynamic system or an expected value in a static system. In systems that evolve over time, we can simulate the operation of the system for, say, n time units and then compute an appropriate performance measure. In static systems we can repeatedly sample instances of the system and compute an average.

To be more precise, many problems in simulation-based optimization can be modelled by a real (or vector)-valued stochastic process $\{f_n(x, \omega) \mid n = 1, 2, \ldots\}$. For each $n \geq 1$ and each $x \in \mathbf{R}^k$, $f_n(x, \omega)$ are random variables defined on a common probability space $(\Omega, \mathcal{F}, P)$. It is helpful to keep in mind that x represents the decision variable, ω represents the sample path (which is indeed a list random numbers that give rise to a particular simulation run), and n represents the simulation length.

We assume the existence of a limit function $f_\infty(x, \omega)$ such that for every x, $f_n(x, \omega) \to f_\infty(x, \omega)$ as $n \to \infty$ at almost every $\omega \in \Omega$. We are interested in finding a minimizer of f_∞ and in general we can only observe f_n for finite n. Therefore we will approximate minimizers of f_∞ using such information about f_n. The method is simple: fix an $\omega \in \Omega$ (by the method of common random numbers) and a large n (to get a good estimate of the limit function), compute a minimizer $x_n^*(\omega)$ of $f_n(\cdot, \omega)$, and take $x_n^*(\omega)$ as an approximate minimizer of $f_\infty(\cdot, \omega)$. Note that minimizers of $f_\infty(x, \omega)$ may generally depend on the sample path ω. However, in many practical problems for which one would anticipate using this technique, f_∞ is a deterministic function, for example a steady-state performance function or an expectation, i.e. it is independent of ω.

The basic case of sample-path optimization, concerning the solution of simulation optimization problems with deterministic constraints, appeared in Plambeck et al. (1993, 1996) and was analyzed in Robinson (1996). Robinson (1996) gave a number of sufficient conditions (on $\{f_n(x, \omega)\}$) which guarantee that if we take n large enough, each approximate minimizer $x_n^*(\omega)$ will be close to a true minimizer of the limit function. Plambeck et al. (1993, 1996) reported extensive numerical experiments on large scale systems (project management involving PERT networks with up to 110 stochastic arcs and cycle time optimization in unreliable tandem production lines with up to 50 machines). In turn, Gürkan (2000) focused on optimization of buffer allocations in unreliable tandem production lines and reported results also for systems with up to 50 machines. Both Plambeck et al. (1993, 1996) and Gürkan (2000) used IPA for gradient estimation.

In the static case, a closely related technique centered around likelihood-ratio methods appeared in Rubinstein and Shapiro (1993) under the name of stochastic counterpart methods. The basic approach (and its variants) is also known as sample average approximation method in the stochastic programming literature; see for example Shapiro and Homem-De-Mello (1998) and Linderoth et al. (2002) for some of the recent progress in the static context of stochastic programming.

In Gürkan et al. (1996, 1999a) the basic idea of using sample-path information was extended to solving stochastic variational inequalities and equilibrium problems. This work was used further in Gürkan et al. (1999b) for establishing almost-sure convergence of sample-path methods when dealing with general stochastic optimization problems with stochastic constraints. More recently, Birbil et al. (2006) deals with the theoretical analysis and application of sample-path methods to the so-called stochastic mathematical programs with equilibrium constraints.

In this paper, we do not deal with theoretical issues related to convergence analysis; instead we solely focus on the operational issues that need to be addressed in solving the hedging point problem in practise. In general, doing a rigorous convergence analysis would require one to have additional insight to path-wise functional properties of the performance measures involved. This can be achieved if one is willing to make additional assumptions, say focus on certain distribution functions, as illustrated in Gürkan (2000). There, certain path-wise functional properties of sample throughput (as function of buffer capacities) were used to verify the sufficient conditions established in Robinson (1996) and almost sure convergence of sample-path optimization was proven. Here, we would like to work in full generality and we only verify the convergence of our procedure through numerical experiments.

There are two key points that make sample-path type methods attractive in practise: (1) once we fix a sample point ω and n, $f_n(x, \omega)$ becomes a deterministic function of x; (2) IPA—when it is applicable—is able to compute *exact* gradients of $f_n(x, \omega)$. With these observations, very powerful methods of constrained and unconstrained deterministic optimization become available for use on the $f_n(\cdot, \omega)$. In the smooth case we can apply superlinearly convergent methods like the BFGS algorithm (or a variant for constrained problems) to minimize f_n to high accuracy in relatively few function and gradient evaluations. For more information on these algorithms see Fletcher (1987) and Gill et al. (1981), and for the software available see Moré and Wright (1993). Using superlinearly convergent methods enables us to be confident about the location and the accuracy of the minimizer of f_n, because we can differentiate between the errors due to the approximation of f_∞ by f_n and those due to the inaccurate computation of a minimizer of f_n. With slower algorithms like stochastic approximation, this is difficult if not impossible.

It is well-known that in many cases in which one uses a discrete event simulation, the exact partial derivatives (or directional derivatives) of $f_n(x, \omega)$ with respect to x can be obtained by using IPA in a single simulation run; see Glasserman (1991) and Ho and Cao (1991). To this end, we propose developing a GSMP representation of the system and utilizing that representation to derive recursive expressions for the exact derivatives. A GSMP can simply be thought as a mathematical framework which models the evolution of a discrete-event simulation. As it can be seen in Section 4, the GSMP framework enables us to develop an intuitive and transparent algorithm for a general system with multiple machine and demand states and with state transitions that are governed by general continuous distributions.

Finally, we would like to point out that our general approach, with appropriate modifications, could be applicable to several other challenging problems encountered in practise, especially considering the recent trends in using stochastic fluid models for decision making under uncertainty; see for example Liu and Gong (2002) and Sun et al. (2004).

3 Problem description

Consider a manufacturing system that produces a single product. The maximum production capacity of the system can take one of K possible values, r_i $(i = 1, 2, \ldots, K)$ according to a stationary process. We denote by $\alpha(t)(\alpha(t) = 1, 2, \ldots, K)$ the state of the machine at time t and $r_{\alpha(t)}$ the corresponding maximum production capacity in that state. The time spent in state i is a continuous random variable with a general distribution F_i. After state i, the next machine state k, is determined according to a matrix $\{P_{ik}\}$ with $\sum_{k=1}^{K} P_{ik} = 1$.

The system has to cope with a random demand. Similar to machine states, we denote by $\beta(t)(\beta(t) = 1, 2, \ldots, M)$ the state of demand at time t and $d_{\beta(t)}$ the corresponding demand rate in that state. The time spent in demand state j also has a general distribution G_j and after state j, the next demand state k is determined according to a matrix $\{Q_{jk}\}$ with $\sum_{k=1}^{M} Q_{jk} = 1$. Note that although in many interesting problems occurring in practise $P_{ii} = Q_{jj} = 0$ would hold, we don't necessarily require it in our model.

Let $X(t)$ denote the inventory level at time t. Note that $X(t)$ can take negative values corresponding to backorders in the system. To this end, it is useful to distinguish the positive and negative parts of the inventory process. We define by $X^+(t) = X(t)I_{\{X(t)\geq 0\}}$ the production surplus of the system and by $X^-(t) = -X(t)I_{\{X(t)<0\}}$ the backorder level (where I_A is the indicator function corresponding to set A). The instantaneous cost function is then given as: $c^+X^+(t) + c^-X^-(t)$ where c^+ and c^- are respectively the holding and backorder costs per item per unit time.

We can classify the combined states of the system as (i, j) $(i = 1, 2, \ldots, K, j = 1, 2, \ldots, M)$ considering the maximum production rates and demand rates simultaneously. A state (i, j) is called deficient if $r_i < d_j$. In such a state $X(t)$ has to be decreasing regardless of the production control selected. Similarly, state (i, j) is called non-deficient if $r_i \geq d_j$; in this case $X(t)$ can be increasing, decreasing, or constant depending on the actual production control. Actually, if $r_i = d_j$, $X(t)$ can be either decreasing or constant and such a state is called a zero state.

Let $v(t)$ $(r_{\alpha(t)} \geq v(t) \geq 0)$ denote the actual production rate at time t. The objective of the manufacturing flow control problem is to select $v(t)$ such that

$$\lim_{T\to\infty} \frac{1}{T} \int_0^T c^+X^+(t) + c^-X^-(t)dt \tag{3.1}$$

is minimized. We study the above problem with the restriction that $v(t)$ is in the class of hedging point policies. A hedging point policy drives the system to the hedging point of the current system state at the fastest rate possible and then keeps it at the hedging point by adjusting the production rate until a system state change occurs. Accordingly, for each non-deficient system state (i, j), there is a hedging point z_{ij}. The control $v(t)$ is completely determined by the specification of z_{ij}'s for non-deficient machine states:

$$v(t) = \begin{cases} r_i & \text{if } X(t) < z_{ij} \\ d_j & \text{if } X(t) = z_{ij} \\ 0 & \text{if } X(t) > z_{ij} \end{cases}$$

and in convention with the existing literature (see Liberopoulos and Hu (1995) for example) we assume that the production rate in deficient states is set to the maximum rate (i.e. $v(t) = r_i$) regardless of the inventory position. Naturally, other production rules can also be easily formulated within the same framework. The control problem under the restriction to this particular class of policies is to select the optimal values of the hedging points z_{ij} in order to minimize expression (3.1).

An important issue is the stability of such a system. Let π_i ($i = 1, 2, \ldots, K$) denote the stationary probability of being in machine state i and q_j ($j = 1, 2, \ldots, M$) denote the stationary probability distribution of being in demand state j. Then a sufficient condition for stability of the system is given by:

$$\sum_{i=1}^{K} \pi_i r_i > \sum_{j=1}^{M} q_j d_j. \tag{3.2}$$

We provide a proof of this in Appendix 1. Hereon, we will restrict our attention to systems satisfying condition (3.2).

In addition, we assume that the process $X(t)$ does not "hit and run" any of the hedging points. That is, there exist an $\epsilon > 0$ such that if $t^{'}$ is a time that $X(t^{'})$ becomes z_{ij}, then we should have $X(t^{'} + \epsilon) = z_{ij}$ for each i and j. Following the terminology of Brémaud et al. (1997), we will call this assumption "hit and stick." "Hit and stick" is almost always satisfied: In practise, it means that two events (end of holding time in a state and reaching to a hedging level) do not happen at the same instant which is satisfied almost surely, since the probability of a continuous variable being equal to a specific value is always 0. Roughly speaking, excluding the possibility of the simultaneous occurrence of exogenous (end of holding time in a state) and endogenous (reaching to a hedging level) events helps to ensure the differentiability of the performance measure. Otherwise, one can still develop expressions for one-sided directional derivatives, but they may not be equal, as also noted by Brémaud et al. (1997) for a two hedging point system.

4 The GSMP and gradient estimation

As mentioned earlier, to carry out the subsequent simulation and infinitesimal perturbation analysis, we will utilize a generalized semi-Markov process (GMSP) representation of the system. Glynn (1989) provides a brief introduction and an excellent description of this framework can be found in Shedler (1993).

The basic idea of a GSMP can be explained as follows: There is a set of states and a set of events. The GSMP jumps from one state to another upon the occurrence of an event; at each state there are some active events. At any time, each active event is associated with a clock representing the residual lifetime of that event and a speed at which the clock runs down. If the clock corresponding to event e in state s equals k and the speed at which this clock runs is r, then e is scheduled to occur after k/r units of time. The next event and the time until it occurs are always determined by the smallest clock reading/clock speed ratio. More formally, let $k(e, t)$ be the reading of the clock for event e at time t and $r(e, t)$ be the speed at which that clock runs down. $E(t)$ is the current set of active events, i.e. the set of events with $r(e, t) \neq 0$. Then the next event to occur is given by

$$e^*(t) = \operatorname{argmin} \left\{ \frac{k(e,t)}{r(e,t)} | e \in E(t) \right\}.$$

Upon the occurrence of an event, changes may occur in the physical state, clock settings, and clock speeds: If event e occurs in state s, the process may move to a new state s' with a certain probability $p(s'; s, e)$; the set of active events changes with the state; clocks for any old events which remain active continue to run in the new state; new clocks are initialized for all new active events and for the event which just occurred if it is also active in the new state. The initial value of each new clock for event e in state s is a random variable with a pre-specified cumulative distribution function $F(\cdot; s, e)$. This goes on until a termination criteria is reached.

To develop a GSMP representation for our system, let the current system states be $\alpha(t) = i$ and $\beta(t) = j$ and then define the following events:

$\mathcal{MF}$: The end of a sojourn time at current machine state i,
$\mathcal{UH}$: Reaching the hedging point z_{ij} of current system state (i, j) from below,
$\mathcal{DH}$: Reaching the hedging point z_{ij} of current system state (i, j) from above,
$\mathcal{DG}$: The end of a sojourn time at current demand state j,
$\mathcal{T}_f$: The simulation termination.

We let $u(t) = v(t) - d_{\beta(t)}$ to denote the fill-in rate, the total rate of change in the inventory level process, at time t. Also let $W^M(t)$ denote the remaining time until the current machine state changes and $W^D(t)$ denote the remaining time until the current demand state changes. If we let T to denote the prespecified amount of time that the system will be simulated, we can then define the clock readings and associated speeds as in Table 1.

Next, utilizing this GSMP representation, we derive a recursive formula to compute *exact* derivatives of long-run average cost per unit time with respect to hedging points, in a single simulation run. As mentioned earlier, we use IPA to compute derivatives of long-run average cost per unit time. Let $f : \mathbf{R}^{KM} \to \mathbf{R}$ be a function which has a gradient at a point $z = (z_1, \ldots, z_{KM})$, then IPA computes an array whose (i, j)th component is

$$\frac{df}{dz_{ij}} = \lim_{\Delta z \to 0} \frac{f(z + \Delta z y_{ij}) - f(z)}{\Delta z}$$

where $y_{ij} (i = 1, \ldots, K, j = 1, \ldots, M)$ is the (i, j)th unit vector in $\mathbf{R}^{KM}$. In the following we use $d(\cdot)/dz_{ij}$ to denote the (i, j)th partial derivative of $(\cdot)$ at the point z.

Table 1 The clock readings and associated speeds

e	$\mathcal{MF}$	$\mathcal{UH}$	$\mathcal{DH}$	$\mathcal{DG}$	$\mathcal{T}_f$
$k(e,t)$	$W^M(t)$	$z_{ij} - X(t)$	$X(t) - z_{ij}$	$W^D(t)$	$T - t$
$r(e,t)$	1	$u(t)$	$-u(t)$	1	1

Furthermore, we abuse notation and use $z + \Delta z_{ij}$ and $z + \Delta z y_{ij}$ interchangeably; that is $\Delta z_{ij} = \Delta z y_{ij}$.

When $z = (z_{ij})$ is the vector of hedging points, as T gets large, the long-run average cost per unit time could be defined as $J_T(z) = C(T)/T$, where $C(T)$ is the cumulative cost incurred by the system over time T as given by

$$C(T) = \int_0^T c^+ X^+(t) + c^- X^-(t) dt. \tag{4.3}$$

Let $t_0, t_1, \ldots$ be the event occurrence times in a sample path and $\tau_n = t_n - t_{n-1}$ be the time between the $(n-1)$st and nth events. Without loss of generality we assume $t_0 = 0$. If the nth event is the event that the system has been working T time units, i.e. $t_n = T$ for some n, we have

$$\frac{dJ_T(z)}{dz_{ij}} = \frac{d}{dz_{ij}} \left(\frac{C(T)}{T} \right) = \frac{dC(T)}{dz_{ij}} \frac{1}{T} = \frac{dC(t_n)}{dz_{ij}} \frac{1}{t_n}. \tag{4.4}$$

From Eq. 4.4 we see that $J_T(z)$ has derivatives if and only if $C(t_n)$ has them. Next we show that $C(t_n)$ has the desired property and derive a recursive expression for $dC(t_n)/dz_{ij}$, which is a quantity computable from the simulation information generated up time t_n.

For $n = 0, 1, \ldots$ let e_{n+1} be the $(n+1)st$ event; then

$$e_{n+1} = e^*(t_n) = \text{argmin} \left\{ \frac{k(e, t_n)}{r(e, t_n)} | e \in E(t_n) \right\},$$

and

$$t_{n+1} = t_n + \tau_{n+1} \qquad \text{where} \qquad \tau_{n+1} = \frac{k(e_{n+1}, t_n)}{r(e_{n+1}, t_n)}. \tag{4.5}$$

Recall that $v(t_n)$ is the actual production rate and $u(t_n)$ is the fill-in rate at t_n; that is, if the controller is in state (i, j) at t_n, then $u(t_n) = v(t_n) - d_j$. Equation 4.6 below gives a recursive expression for $C(t_{n+1})$.

$$\begin{aligned} C(t_{n+1}) = {} & C(t_n) \\ & + I_{\{X(t_n) \geq 0\}} I_{\{X(t_{n+1}) \geq 0\}} c^+ \left[X(t_{n+1}) \tau_{n+1} - \frac{u(t_n) \tau_{n+1}}{2} \tau_{n+1} \right] \\ & + I_{\{X(t_n) < 0\}} I_{\{X(t_{n+1}) < 0\}} c^- \left[-X(t_{n+1}) \tau_{n+1} + \frac{u(t_n) \tau_{n+1}}{2} \tau_{n+1} \right] \\ & + I_{\{X(t_n) \geq 0\}} I_{\{X(t_{n+1}) < 0\}} \left\{ c^+ \left[\frac{-X(t_n)}{2} \frac{X(t_n)}{u(t_n)} \right] + c^- \left[\frac{-X(t_{n+1})}{2} \frac{X(t_{n+1})}{u(t_n)} \right] \right\} \\ & + I_{\{X(t_{n+1}) \geq 0\}} I_{\{X(t_n) < 0\}} \left\{ c^+ \left[\frac{X(t_{n+1})}{2} \frac{X(t_{n+1})}{u(t_n)} \right] + c^- \left[\frac{X(t_n)}{2} \frac{X(t_n)}{u(t_n)} \right] \right\}. \end{aligned} \tag{4.6}$$

Here is the main theorem.

Theorem 1 *For $n \geq 0$, $C(t_{n+1})$ has partial derivatives at z for all z and $i = 1, \dots, K$, $j = 1, \dots, M$, given by*

$$\begin{aligned}\frac{dC(t_{n+1})}{dz_{ij}} &= \frac{dC(t_n)}{dz_{ij}} \\ &\quad + I_{\{X(t_n)\geq 0\}} I_{\{X(t_{n+1})\geq 0\}} c^+ \left[\frac{dX(t_{n+1})}{dz_{ij}} \tau_{n+1} + \frac{d\tau_{n+1}}{dz_{ij}} X(t_{n+1}) - u(t_n)\tau_{n+1} \frac{d\tau_{n+1}}{dz_{ij}} \right] \\ &\quad + I_{\{X(t_n)<0\}} I_{\{X(t_{n+1})<0\}} c^- \left[- \frac{dX(t_{n+1})}{dz_{ij}} \tau_{n+1} - \frac{d\tau_{n+1}}{dz_{ij}} X(t_{n+1}) + u(t_n)\tau_{n+1} \frac{d\tau_{n+1}}{dz_{ij}} \right] \\ &\quad + I_{\{X(t_n)\geq 0\}} I_{\{X(t_{n+1})<0\}} \left\{ c^+ \left[\frac{-X(t_n)}{u(t_n)} \frac{dX(t_n)}{dz_{ij}} \right] + c^- \left[\frac{-X(t_{n+1})}{u(t_n)} \frac{dX(t_{n+1})}{dz_{ij}} \right] \right\} \\ &\quad + I_{\{X(t_{n+1})\geq 0\}} I_{\{X(t_n)<0\}} \left\{ c^+ \left[\frac{X(t_{n+1})}{u(t_n)} \frac{dX(t_{n+1})}{dz_{ij}} \right] + c^- \left[\frac{X(t_n)}{u(t_n)} \frac{dX(t_n)}{dz_{ij}} \right] \right\}. \end{aligned} \tag{4.7}$$

The proof of Theorem 1 is by induction. In practise, to compute $dC(t_{n+1})/dz_{ij}$ we need to be able to compute $dX(t_{n+1})/dz_{ij}$ (done in Lemma 3 below), $d\tau_{n+1}/dz_{ij}$ (done in Lemmas 1, 4, 5, 6 below), and $du(t_n)/dz_{ij}$ (done in Lemma 2 below) recursively, using information that is available up to t_{n+1}. Before proceeding with the proof of the theorem, we first mention the concept of "similarity" (which we address in detail in Appendix 2) and state a few technical lemmas. The proofs of all lemmas of this section are in Appendix 3.

"Similarity" of the nominal path and the perturbed path (which are made precise in Appendix 2) is a standard issue one needs to deal with when developing IPA algorithms. In Appendix 2, we show that along any sample path of finite length, say k events, with $min\{\tau_n | n = 1, 2, \dots, k\} > 0$, there is always a $\Delta z_{ij} > 0$ (or $\Delta z_{ij} < 0$ whose size depends on the sample path) small enough such that increasing (or decreasing, respectively) z by Δz_{ij} does not cause any event order change; that is, the perturbed path and the nominal path remain similar.

Lemma 1 *For all $n = 0, 1, \dots$, $r(e_{n+1}, t_n)$ has partial derivatives at z for all z and $i = 1, \dots, K$, $j = 1, \dots, M$, which are* 0.

Lemma 2 *For all $n = 0, 1, \dots$, $u(t_n)$ has partial derivatives at z for all z and $i = 1, \dots, K$, $j = 1, \dots, M$, which are* 0.

Lemma 3 *Suppose that $X(t_{n-1})$ and τ_n have partial derivatives at z for all z and $i = 1, \dots, K$, $j = 1, \dots, M$. Then $X(t_n)$ has partial derivatives at z for all z and $i = 1, \dots, K$, $j = 1, \dots, M$ given by*

$$\frac{dX(t_n)}{dz_{ij}} = \frac{dX(t_{n-1})}{dz_{ij}} + u(t_{n-1}) \cdot \frac{d\tau_n}{dz_{ij}}.$$

Lemma 4 *Suppose that $W^M(t_{n-1})$, $W^D(t_{n-1})$, and τ_n have partial derivatives at z for all z and $i = 1, \ldots, K$, $j = 1, \ldots, M$. Then $W^M(t_n)$ and $W^D(t_n)$ have partial derivatives at z for all z and $i = 1, \ldots, K$, $j = 1, \ldots, M$ given by*

$$\frac{dW^M(t_n)}{dz_{ij}} = \frac{dW^M(t_{n-1})}{dz_{ij}} - \frac{d\tau_n}{dz_{ij}}.$$

and

$$\frac{dW^D(t_n)}{dz_{ij}} = \frac{dW^D(t_{n-1})}{dz_{ij}} - \frac{d\tau_n}{dz_{ij}}.$$

Lemma 5 *Suppose that $X(t_n)$, $W^M(t_n)$, $W^D(t_n)$, and t_n have partial derivatives at z for all z and $i = 1, \ldots, K$, $j = 1, \ldots, M$. Then $k(e_{n+1}, t_n)$ has partial derivatives at z for all z and $i = 1, \ldots, K$, $j = 1, \ldots, M$ given by*

$$\frac{dk(e_{n+1}, t_n)}{dz_{ij}} = \begin{cases} \dfrac{dW^M(t_n)}{dz_{ij}} & \text{if } e_{n+1} = \mathcal{MF}, \\ I_{\{i=\alpha(t_n),\, j=\beta(t_n)\}} - \dfrac{dX(t_n)}{dz_{ij}} & \text{if } e_{n+1} = \mathcal{UH}, \\ \dfrac{dX(t_n)}{dz_{ij}} - I_{\{i=\alpha(t_n),\, j=\beta(t_n)\}} & \text{if } e_{n+1} = \mathcal{DH}, \\ \dfrac{dW^D(t_n)}{dz_{ij}} & \text{if } e_{n+1} = \mathcal{DG}, \\ -\dfrac{dt_n}{dz_{ij}} & \text{if } e_{n+1} = \mathcal{T}_f. \end{cases} \tag{4.8}$$

Lemma 6 *Suppose that t_n has partial derivatives at z for all z and $i = 1, \ldots, K$, $j = 1, \ldots, M$. Then t_{n+1} has partial derivatives at z for all z and $i = 1, \ldots, K$, $j = 1, \ldots, M$ given by*

$$\frac{dt_{n+1}}{dz_{ij}} = \frac{dt_n}{dz_{ij}} + \frac{d\tau_{n+1}}{dz_{ij}} \tag{4.9}$$

where

$$\frac{d\tau_{n+1}}{dz_{ij}} = \frac{1}{r(e_{n+1}, t_n)} \frac{dk(e_{n+1}, t_n)}{dz_{ij}}. \tag{4.10}$$

The next lemma will start the induction.

Lemma 7 *t_1, τ_1, $X(t_0)$, $W^M(t_0)$, $W^D(t_0)$, and $C(t_0)$ have partial derivatives at z for all z and $i = 1, \ldots, K$, $j = 1, \ldots, M$. These are given by, for all i and j:*

$$\frac{dX(t_0)}{dz_{ij}} = \frac{dW^M(t_0)}{dz_{ij}} = \frac{dW^D(t_0)}{dz_{ij}} = \frac{dC(t_0)}{dz_{ij}} = 0 \text{ and}$$

$$\frac{dt_1}{dz_{ij}} = \frac{d\tau_1}{dz_{ij}} = \frac{1}{r(e_1,t_0)}\frac{dk(e_1,t_0)}{dz_{ij}} = \begin{cases} \frac{1}{u(t_0)} I_{\{i=\alpha(t_0),\, j=\beta(t_0)\}} & \textit{if } \; e_1 = \mathcal{UH}, \\ \frac{1}{u(t_0)} I_{\{i=\alpha(t_0),\, j=\beta(t_0)\}} & \textit{if } e_1 = \mathcal{DH}, \\ 0 & \textit{otherwise.} \end{cases}$$

Now we are ready to prove the main theorem.

Proof of Theorem 1 First, observe that because of Remark 1 (in Appendix 2), we can treat $I_{\{X(t_n(z))\geq 0\}}$, $I_{\{X(t_{n+1}(z))\geq 0\}}$, $I_{\{X(t_n(z))<0\}}$, and $I_{\{X(t_{n+1}(z))<0\}}$ as constants while using the chain rule of differentiation. Next, Lemma 7 provides the start of inductive argument. Suppose that t_n, τ_n, $X(t_{n-1})$, $W^M(t_{n-1})$, $W^D(t_{n-1})$, and $C(t_{n-1})$ have partial derivatives at z for all z and $i = 1, \ldots, K$, $j = 1, \ldots, M$. Apply Lemma 4 to get the recursions for $W^M(t_n)$ and $W^D(t_n)$; Lemma 3 for $X(t_n)$; and Lemma 6 for t_{n+1} and τ_{n+1}. Lemmas 1 and 5 provide the necessary information about $dk(e_{n+1}, t_n)/dz_{ij}$ and $dr(e_{n+1}, t_n)/dz_{ij}$. □

In Appendix 4, we provide a pseudo code for our simulation and IPA algorithm that we developed utilizing our GSMP representation and the recursions developed in this chapter. The steps (needed for the IPA algorithm) added to the basic simulation algorithm are marked as “IPA”.

Note that the GSMP framework together with the recursive manner of computing derivatives allows one to use IPA also for computation of second-order derivatives. However, we have not pursued this here, since most of the state-of-art deterministic nonlinear optimization codes (including E04UCC we use in our numerical experiments reported in Section 5) are capable of using just the derivative information for maintaining and utilizing efficiently dynamic BFGS quasi-Newton approximations for the Hessian.

It is also worth noting that an important issue in gradient estimation literature has been the development of estimators with good asymptotic behavior. The convergence theorems and convergence rate results for stochastic approximation type methods are concerned with the unbiasedness of the gradient estimator, for example see Kushner and Clark (1978); whereas convergence theorems for sample-path optimization may not have these requirements in general, for example see Robinson (1996). When using sample-path optimization the main requirement from the practical point of view is an *exact* gradient or a directional derivative (whichever is available) of the sample function to be optimized so that an efficient deterministic optimizer is able to utilize this sensitivity information. In this sense, the sample-path performance functions (for finite time) of the buffer optimization problem addressed in Gürkan (2000) constitute an interesting example. They were discontinuous functions and their exact derivatives were most likely biased. However, under the assumption that the steady-state throughput is a continuous function of buffer capacities, Gürkan (2000) proved the almost sure convergence of sample-path optimization. Furthermore, the deterministic optimizer was able to utilize the *exact* directional derivatives of the discontinuous sample functions efficiently and find the optimal buffer allocations in systems with up to 50 unreliable machines.

Proving the unbiasedness of the estimator we derived here is rather standard and we tackle that next.

Lemma 8 $|\frac{J_T(z+\Delta z_{ij},\omega)-J_T(z,\omega)}{\Delta z_{ij}}| \leq (c^+ + c^-)$*, for any* Δz_{ij} *small enough so that the event order list does not change, where* ω *is the sequence of random numbers which give rise to a particular simulation run.*

Theorem 2 *IPA provides an unbiased gradient estimator; that is, we have* $E[\frac{dJ_T(z,\omega)}{dz_{ij}}] = \frac{dE[J_T(z,\omega)]}{dz_{ij}}$.

Proof From Lemma 8, we have $|\frac{J_T(z+\Delta z_{ij},\omega)-J_T(z,\omega)}{\Delta z_{ij}}| \leq (c^+ + c^-)$. From Theorem 1,

$$\frac{dJ_T(z,\omega)}{dz_{ij}} = \lim_{\Delta z_{ij}\to 0} \frac{J_T(z+\Delta z_{ij},\omega)-J_T(z,\omega)}{\Delta z_{ij}}$$

exists at any z with probability 1.

We choose $g(\omega) := (c^+ + c^-)$ as the dominating function and use Lebesgue's dominated convergence theorem to obtain

$$\begin{aligned}\frac{dE[J_T(z,\omega)]}{dz_{ij}} &= \lim_{\Delta z_{ij}\to 0} \int_\Omega \frac{J_T(z+\Delta z_{ij},\omega)-J_T(z,\omega)}{\Delta z_{ij}} dP \\ &= \int_\Omega \lim_{\Delta z_{ij}\to 0} \frac{J_T(z+\Delta z_{ij},\omega)-J_T(z,\omega)}{\Delta z_{ij}} dP \\ &= E\left[\frac{dJ_T(z,\omega)}{dz_{ij}}\right].\end{aligned}$$

□

5 Numerical experiments

In this section, we report some of our numerical experiments used for testing the empirical performance of the ongoing methodology. First, we compared the analytical results of some production control problems found in literature with our results. Then, we solved five different examples with dimensions up to ten machine and four demand states. In Example 1, we also compared sample-path optimization (SPO) with a variant of stochastic approximation (SA), called single-run optimization (SRO) (see Meketon 1987), since SRO variant of SA is reported to perform better than classical SA for some problems in the literature; see for example Leung (1990). In all numerical experiments, we used the exponential distribution for state transition times. Essentially any continuous distribution whose support is on $(0,\infty)$ can be used; the choice of the exponential distribution is mainly made for comparison and convenience.

In order to simulate a general model with K machine and M demand states, we use our pseudo code written in C which calculates the gradients and the value of the objective function and constraints in a single simulation run. We convey these values to the optimizer which determines an "optimal" point. As the optimizer, we use the deterministic nonlinear optimization code E04UCC of NAG C library, Mark 7, NAG (2002). E04UCC is designed to minimize an arbitrary smooth function subject to constraints, which may include simple bounds on the variables, linear constraints, and smooth nonlinear constraints. Essentially, it is a sequential quadratic programming method incorporating an augmented Lagrangian merit function and a BFGS quasi-Newton approximation to the Hessian of the Lagrangian. The code iteratively determines the total number of simulation runs N required to find an approximate minimizer; this is controlled by the "Optimality Tolerance ($OptTol$)." The parameter $OptTol$ specifies the accuracy to which the user wishes the final iterate to approximate a solution of the problem. Broadly speaking, $OptTol$ indicates the number of correct figures desired in the objective function at the solution. For example, if $OptTol$ is 10^{-6}, the final value of the objective function would have approximately six correct figures. When there are only linear constraints, E04UCC considers a point "optimal" if the current step length, the norm of the search direction, and the norm of the projected gradient become sufficiently small. In the presence of nonlinear constraints, to be considered "optimal," the point should also satisfy an additional stopping criteria; namely, the violation of all active constraints should also become sufficiently small, see NAG (2002) for additional details. It is worth noting that in the absence of convexity of the functions involved, a nonlinear optimization program like E04UCC (rather than a specialized global optimization code) can only guarantee to find a local optimum which may or may not be a global optimum.

As mentioned earlier, to verify the correctness and accuracy of our numerical procedures, we first compared our results with other results we could find in the literature. These include analytical results reported in Bielecki and Kumar (1988) and Gershwin (1994), and analytical/numerical results reported in Haurie et al. (1994) (their single-part model), Yan et al. (1994) (their single-stage model), and Liberopoulos and Caramanis (1994) (their single-part model). Aside from the system of Liberopoulos and Caramanis (1994) which had two hedging points, all the others were systems with a single hedging point. In all cases, we observed that our results converged to the correct optimal solutions with relatively small computational effort.

Additional test cases are provided by Liberopoulos and Hu (1995). They analytically established the relative ordering of optimal hedging points of more than two machine states in four specific Markov chain systems. In our experiments, we used their Systems 1, 2, and 3 with different numbers of machine states and parameter values. Optimal hedging points that we found had always the same ordering properties established by Liberopoulos and Hu (1995).

All the runs reported here are performed by a Dell PC with Dual-Intel Xeon, 2.66 GHz processors, and 2 GB 266 MHz DDR Non-ECC SDRAM Memory using Windows 2000.

5.1 Optimization of unconstrained production control problems

In this part, we report the results of three numerical experiments. An additional test for the quality of the solutions we compute in these experiments is provided through a well-known result: the probability that the system is in backlog state ($P(X(t) < 0)$)

Table 2 Specifications of 4-machine state system with constant demand rate for Example 1

Machine states	1	2	3	4	d	c^+	c^-	$c^+/(c^+ + c^-)$
Maximum production rate	0	4	80	40	9.533	1	10	0.090910

at an optimal solution should be equal to the proportion $c^+/(c^+ + c^-)$. In our computations, we approximate the probability that the system is in backlog state by computing the long run proportion of time that the system is in backlog state which we denote by $P_{blog}(z)$.

Example 1 The first experiment is with four machine states and a constant demand rate. Table 2 contains the data of this problem and the corresponding transition matrix of the four machine states is in Fig. 1.

In this example, we have two non-deficient machine states, namely 3 and 4. We used $T = 500,000$ simulation time per simulation call which corresponded to around $5,000,000$ machine events in order to get a good estimate of the limit function. We set $OptTol$ to 10^{-6} and solved the problem from five different initial points A, B, C, D, and E; we observed convergence to the same final point z^*_T. As we do not know the analytical solution, we found the "optimal" solution z^*_∞ and the function values $J_\infty(z)$ by using SPO with a very large computational effort; using a simulation time of $5,000,000$ per simulation call which corresponded to around 50,000,000 machine events. The results are summarized in Table 3. Recall that N is the number of simulation runs determined by E04UCC to find an approximate minimizer.

As mentioned earlier, for comparison, we also solved this example by using SRO algorithm of the form: $z_{n+1} = \Pi_\Phi(z_n - \frac{a_0}{n} g_n)$, where z_n is the hedging point vector at the nth iteration, g_n is the gradient estimate at z_n, a_0 is the predetermined step size constant, and $\Pi_\Phi(\cdot)$ is the projection onto the feasible set Φ determined by the simple bound constraints. It is well-known that it is possible for an SRO algorithm to diverge if one does not impose explicit bounds on the variables to ensure the boundedness of the iterates. We used a lower bound with each element equal to zero and an upper bound with each element equal to 1,000.

We tried to observe the behavior of SRO iterates under the same computational budget used by SPO. In our implementation of SRO, for each initial point, we used $T = 100,000$ simulation time per simulation call which corresponded to around $1,000,000$ machine events and we made $5N$ iterations (i.e. simulation runs), each run corresponding to one iteration of the algorithm. We chose to have a runlength of approximately $1,000,000$ machine events per iteration since Liberopoulos and Caramanis (1994) used this value in their numerical experiments with the stochastic approximation method. Since the simulation time T used in each iteration of SRO is one fifth of the simulation time T used in each iteration of SPO, we made $5N$ iterations for SRO so that we could compare both methods under the same computational

Fig. 1 Transition rate matrix of machine states for Example 1

$$\begin{bmatrix} -4 & 4 & 0 & 0 \\ 0 & -8 & 0 & 8 \\ 14 & 0 & -15 & 1 \\ 0 & 1 & 9 & -10 \end{bmatrix}$$

Table 3 Optimal solution and the run-time information for Example 1

Machine states	3	4	$J_T(z)$	$P_{blog}(z)$	$J_\infty(z)$
Initial point A	5.00	5.00	8.398	0.176439	8.376
Initial point B	20.00	20.00	17.360	0.001518	17.359
Initial point C	5.00	20.00	11.177	0.063795	11.163
Initial point D	20.00	5.00	12.764	0.021478	12.742
Initial point E	1.00	2.00	18.794	0.576287	18.769
Final point(z_T^*)	7.44	5.27	7.382	0.090910	7.357
"Optimum" point(z_∞^*)	7.45	5.25	7.382	0.090910	7.357

Initial point	N	Total CPU time
A	14	1 min 27 s
B	14	1 min 27 s
C	11	1 min 06 s
D	11	1 min 06 s
E	13	1 min 21 s

budget. Table 4 contains the results of the SRO algorithm with different a_0 values; the *relative error* is calculated as $\left|\frac{J_\infty(z)-J_\infty(z_\infty^*)}{J_\infty(z_\infty^*)}\right|$.

As mentioned earlier, it is well-known that one of the main drawbacks of SA-type algorithms (including SRO variant) is the sensitivity of their empirical performance to the choice of a_0; documented, for example, in Fu and Healy (1992), L'Ecuyer et al. (1994), Glasserman and Tayur (1995), and Gürkan (2000). Our own experiments also confirmed this sensitivity issue even in this small example and we spent considerable effort for finding reasonable values of a_0 which also depend on the initial point. For example, as seen in Table 4, from initial points A, B, C, and D the iterates converge to the optimal solution with a_0 equal to 20; whereas from initial point E, among the ones we tried, the only good value of a_0 is 1.8.

Experimenting with several a_0's was feasible in this example since this was a small problem (with a relative short run time, no constraints, and two decision variables) and we knew where the "optimal" solution was. Since SRO is a first-order gradient method, one would expect to experience the same difficulties more severely in larger examples. Therefore we decided not to compare SRO with SPO in larger unconstrained examples of Section 5.1. For the constrained examples of Section 5.3, since we are not aware of any literature addressing how an SA-type method can handle those, we did not perform comparisons there either.

Example 2 We modified the system in Example 1 by adding four demand states with a transition rate matrix as in Fig. 2. The corresponding demand rates are given in Table 5.

In this example, we have eight non-deficient states and it can be seen that the system copes with a random demand which has an average value equal to the constant demand of Example 1; that is 9.533. We used $T = 1,000,000$ simulation time per simulation call which corresponded to around $12,000,000$ events in order to get a good estimate of the limit function. We again set $OptTol$ to 10^{-6}, solved the problem

Table 4 Solutions generated by SRO with different a_0 values

Initial point	Final point (z_3, z_4) $J_\infty(z)$ *relative error*								Number of iterations (5N)
	$a_0 = 1.5$	$a_0 = 1.8$	$a_0 = 2$	$a_0 = 10$	$a_0 = 15$	$a_0 = 20$	$a_0 = 30$	$a_0 = 40$	
A	(7.13, 5.43) 7.371 0.0019	(7.23, 5.50) 7.366 0.0012	(7.28, 5.53) 7.365 0.0011	(7.45, 5.59) 7.363 0.0008	(7.46, 5.35) 7.358 0.0001	(7.45, 5.27) 7.357 0.0000	(7.45, 5.23) 7.357 0.0000	(7.45, 5.23) 7.357 0.0000	70
B	(15.91, 17.08) 17.786 1.4176	(15.19, 16.44) 13.143 0.7865	(14.72, 16.01) 12.724 0.7295	(7.45, 6.55) 7.432 0.0102	(7.45, 5.28) 7.357 0.0000	(7.45, 5.23) 7.357 0.0000	(7.45, 5.23) 7.357 0.0000	(6.53, 10.37) 8.054 0.0947	70
C	(5.30, 17.64) 10.343 0.4059	(5.36, 17.17) 10.181 0.3838	(5.40, 16.86) 10.075 0.3694	(7.20, 7.78) 7.565 0.0283	(7.46, 5.64) 7.365 0.0011	(7.46, 5.26) 7.357 0.0000	(7.46, 5.24) 7.357 0.0000	(7.47, 5.24) 7.357 0.0000	55
D	(16.12, 4.12) 10.445 0.4197	(15.41, 4.02) 10.074 0.3693	(14.95, 3.97) 9.843 0.3379	(7.48, 5.15) 7.358 0.0001	(7.45, 5.23) 7.357 0.0000	(7.46, 5.24) 7.357 0.0000	(7.33, 7.62) 7.545 0.0255	(3.43, 42.22) 18.713 1.54363	55
E	(7.17, 4.92) 7.372 0.0020	(7.44, 5.21) 7.357 0.0000	(7.59, 5.36) 7.360 0.0004	(13.46, 7.42) 9.618 0.3073	(17.25, 10.21) 12.484 0.6969	(20.78, 14.12) 15.953 1.1684	(27.32, 23.25) 23.422 2.1836	(32.57, 33.84) 30.326 3.1221	65

Fig. 2 Transition rate matrix of demand states for Example 2

$$\begin{bmatrix} -2 & 2 & 0 & 0 \\ 7.5 & -8 & 0.5 & 0 \\ 0 & 0.3 & -3 & -2.7 \\ 0 & 0 & 5 & -5 \end{bmatrix}$$

from different initial points, and observed convergence to the same point. Table 6 contains the summary of the results.

The differences in results of Examples 1 and 2 clearly illustrate that we cannot ignore the stochastic behavior of demand; that is, the variability of demand has an important effect on hedging points. Therefore considering only the average demand without paying attention to the variability of demand leads to incorrect decisions.

Example 3 The third experiment is on a system with ten machine and four demand states. Table 7 contains the data of the problem and the transition rate matrix of machine and demand states are shown in Fig. 3. This system has twenty two non-deficient states. We used $T = 3,000,000$ simulation time per simulation call which corresponded to around $25,000,000$ events.

Here, different from Examples 1 and 2, when we started from different initial points, we observed finding different "optimal" solutions. To make sure that these differences were not caused by insufficient accuracy, we increased $OptTol$ to 10^{-7}, but still observed the same behaviour as reported in Table 8; different initial points are indicated as A, B, and C and different final solutions reached are indicated as z^*_A, z^*_B, and z^*_C, respectively. In general, there could be two major sources for the multiplicity of solutions. First, when an objective function is rather flat near an optimum point (as observed in this example and discussed in Section 5.2.5), there may be several points that can be declared as "optimum" for all practical purposes; a situation termed as "multiple optima." Clearly, all the relevant performance measures would take equal (or very close) values at all of these optimum points. Second, in the absence of (strict) convexity, a function can have several local optima, which may have similar or different performances. We also discuss the non-convex nature of this objective function in more detail in Section 5.2.6. It is worth observing that different solution points in Table 8 have very similar performances.

Furthermore, it can also be seen from Table 8 that some hedging points are close to each other and form clusters. We also discuss this clustering effect in more detail in Section 5.2.7.

5.2 Observations and conjectures from numerical experiments from unconstrained systems

Throughout the numerical experiments, we have done lots of simulation runs and observed some behavior that we found interesting during these runs. Below, we report some of these as observations and conjectures.

Table 5 Demand rates for Example 2

Demand states	1	2	3	4
Rate	5	8	15	20

Table 6 Optimal solution and the run-time information for Example 2

Machine states	3	3	3	3	4	4	4	4	$J_T(z)$	$P_{blog}(z)$
Demand states	1	2	3	4	1	2	3	4		
Initial point A	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	28.766	0.261616
Initial point B	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00	24.390	0.036370
z_T^*	6.72	9.67	24.16	24.78	3.96	7.90	21.23	22.15	17.030	0.090910

Initial point	N	Total CPU time
A	16	7 min 07 s
B	32	13 min 44 s

5.2.1 Effect of load

As mentioned earlier, Liberopoulos and Hu (1995) study ordering properties of optimal hedging points for some special systems. Among those, we used their Systems 1 and 2 with two and three non-deficient and one deficient states to test our simulation runs. After several runs with different parameter settings (with two non-deficient and one deficient states of both System 1 and System 2), we observed that when the load of the system is increased (by either increasing the demand or decreasing the maximum production rates), the proportion of the difference between optimal hedging points to the values of the optimal hedging points themselves decreases. That is, the difference between the optimal hedging points stays constant but because of the increased load, the optimal hedging points increase and the difference between them become negligible compared to their values.

For example, we had an experiment with their System 1 in which there are two hedging points and as the load, ρ, is increased they become close to each other. The transition rate diagram of machine states are given in Fig. 4 and the parameter settings of the system are given in Table 9.

The optimal hedging points of this system are: $z_1^* = 11.5$ and $z_2^* = 8.40$ when the demand rate d is 5 and the corresponding load ρ is 0.44 . When we increase the load by increasing the demand, we obtain the results summarized in Table 10. This leads to the following conjecture:

As the load increases to one, the difference between the optimal hedging points become negligible compared to their values. In other words, the optimal hedging points appear to get close to each other as the load of the system increases.

Table 7 Specifications of 10-machine and 4-demand state system for Example 3

Machine states	1	2	3	4	5	6	7	8	9	10
Maximum production rate	0.4	0	0.8	8	5	1.2	10	6	3	15
Demand states	1	2	3	4	c^+	c^-	$c^+/(c^+ + c^-)$			
Rate	1.3	1.8	3.2	4.0	1	10	0.090910			

Fig. 3 Transition rate diagrams of machine and demand states for Example 3

$$\begin{bmatrix} -12 & 0 & 0 & 4 & 4 & 4 & 0 & 0 & 0 & 0 \\ 0 & -5 & 0 & 0 & 0 & 0 & 2 & 2 & 1 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 7 & 0 & 0 & -8 & 0 & 0 & 0 & 0 & 0 & 1 \\ 4 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 & 0 & 0 & -6 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 & 0 & -5 & 0 & 0 \\ 0 & 0.9 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0.1 \\ 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 2 & -5 \end{bmatrix}$$

$$\begin{bmatrix} -2 & 2 & 0 & 0 \\ 7.9 & -8 & 0.1 & 0 \\ 0 & 0.1 & -5 & 4.9 \\ 0 & 0 & 10 & -10 \end{bmatrix}$$

5.2.2 Irrelevant hedging point

Simulation runs of System 2 of Liberopoulos and Hu (1995) with two non-deficient and one deficient states show that when one of the non-deficient states, say State 1, has a very small long run average maximum production rate compared to the demand rate (i.e. $r_1 > d$ but $r_1\pi_1 \ll d$), the hedging point of State 1 becomes "irrelevant"; that is, the value it takes does not really matter. The optimal hedging point of non-deficient State 2 determines the optimal hedging point of the system. A similar result is obtained with systems having three non-deficient and one deficient states, where $r_1\pi_1 \ll d$, $r_2\pi_2 + r_3\pi_3 > d$, $r_2\pi_2 < d$, and $r_3\pi_3 < d$ hold but it is not the case that $r_2\pi_2 \ll d$ or $r_3\pi_3 \ll d$. In this case, z_1 becomes irrelevant and z_2 and z_3 determine the optimal hedging points of the system. This leads to the following conjecture:

Assume that there is a non-deficient state i which has a very small average maximum production rate compared to the demand rate in the long run, that is $r_i > d$ but $r_i\pi_i \ll d$, but this is not the case for the other non-deficient states in the long run, that is $r_j\pi_j \ll d$ does not hold for $\forall j$, $j \neq i$. For such a system, the optimal hedging point of state i is irrelevant and the optimal hedging points of the system are determined by the optimal hedging points of states j.

5.2.3 Effect of transition diagram and transition probabilities

The more the machine states communicate with each other and the more the state transition probabilities are close to uniform distribution, the more the optimal hedging points get close to each other. In the simulation runs up to systems with 10–15 non-deficient states, we observed that when the machine states form a closed set in which all states communicate with each other with a uniform transition distribution, all of the optimal hedging points are very close to each other. As the transition probabilities between the machine states become less uniform, the optimal hedging points move away from each other.

Table 8 Optimal solution and the run-time information for Example 3

Machine states	Demand states	Initial point A	Initial point B	Initial point C	z_A^*	z_B^*	z_C^*
4	1	5.00	10.00	20.00	19.97	18.30	18.42
4	2	5.00	10.00	20.00	21.68	21.74	22.47
4	3	5.00	10.00	20.00	34.88	29.74	29.72
4	4	5.00	10.00	20.00	29.47	30.03	28.46
5	1	5.00	10.00	20.00	20.10	21.52	21.11
5	2	5.00	10.00	20.00	20.98	22.30	21.95
5	3	5.00	10.00	20.00	29.55	28.89	28.63
5	4	5.00	10.00	20.00	28.82	31.25	33.81
7	1	5.00	10.00	20.00	7.54	7.29	7.33
7	2	5.00	10.00	20.00	9.04	9.00	9.07
7	3	5.00	10.00	20.00	27.65	22.87	21.65
7	4	5.00	10.00	20.00	23.11	21.48	21.38
8	1	5.00	10.00	20.00	7.40	7.35	7.41
8	2	5.00	10.00	20.00	9.49	9.44	10.29
8	3	5.00	10.00	20.00	19.96	21.23	20.45
8	4	5.00	10.00	20.00	23.59	21.42	20.34
9	1	5.00	10.00	20.00	8.31	8.30	8.36
9	2	5.00	10.00	20.00	10.99	12.22	11.45
10	1	5.00	10.00	20.00	13.91	13.70	13.91
10	2	5.00	10.00	20.00	16.03	14.69	13.97
10	3	5.00	10.00	20.00	28.44	28.30	28.17
10	4	5.00	10.00	20.00	27.44	28.42	26.50
$J_T(z)$		28.634	25.116	25.687	22.087	22.085	22.087
$P_{blog}(z)$		0.193072	0.123462	0.057995	0.090841	0.090899	0.090921

Initial point	N	Total CPU time
A	105	2 h 22 min 49 s
B	58	1 h 18 min 47 s
C	55	1 h 15 min 53 s

5.2.4 Effect of demand and production variability

In the simulation runs of Example 3, we have observed that as the production and demand variability increase, the optimal hedging points also increase to hold enough safety stock against the uncertainty in production and demand rates. When the variability in production is not very high, the hedging points get more separated from each other as the demand variability increases. When production variability is high, the hedging points are first separated from each other and then get closer to each

Fig. 4 Transition rate matrix of machine states for example of Section 5.2.1

$$\begin{bmatrix} -2 & 2 & 0 \\ 4 & -5 & 1 \\ 0 & 2 & -2 \end{bmatrix}$$

Table 9 Specifications of example in Section 5.2.1

Machine states	1	2	3	c^+	c^-	$c^+/(c^+ + c^-)$
Maximum production rate	20	40	0	0.1	1	0.090910

other as the demand variability increases. These observations confirm a similar comment made by Tan (2001) for a small system with two machine and two demand states.

5.2.5 Flatness of the objective function and effect of c^+/c^- on flatness

From the simulation runs of Example 3, we have also observed that as the ratio c^+/c^- gets smaller, that is the system has very low holding cost compared to its shortage cost, the objective function value does not change much with considerable increases in hedging points. This is very intuitive since the holding cost is very low, a large increase in the hedging level is reflected to the cost function with a relatively very small effect. Moreover, for a given ratio of c^+/c^-, the objective function is flatter near the optimum points and its flatness increases as the number of hedging points increases.

5.2.6 Non-convexity of the objective function

As also mentioned in Brémaud et al. (1997) the objective function (3.1) is generally not convex in systems with two or more non-deficient machine states. Figure 5a shows the non-convex average cost function with respect to hedging points of machine states 3 and 4 of Example 1; whereas Fig. 5b depicts its level sets. Despite this non-convexity, we reached to the same final solution for Examples 1 and 2 when we started from several different initial points. In comparison, in Example 3 when we started from different initial points, we reached to different final points which is not unexpected when dealing with non-convex functions. As mentioned earlier, in the absence of convexity, a nonlinear optimization program like E04UCC (rather than a specialized global optimization code) can only guarantee to find a local optimum which may or may not be a global optimum. In addition, we should point out that we did not encounter any numerical difficulties in finding the local optimal solutions even in large systems despite the non-convex nature of the objective function. We also note that at different local optimal solutions, the objective function values are very close to each other suggesting also multiple optima.

Table 10 Effect of load for example in Section 5.2.1

d	ρ	z_1^*	z_2^*	$z_1^* - z_2^*$	$\frac{z_1^* - z_2^*}{z_1^*}$
5	0.44	11.50	8.40	3.10	0.27
6	0.52	17.57	14.30	3.27	0.19
7	0.61	26.92	23.37	3.55	0.13
8	0.70	41.96	38.24	3.72	0.09
9	0.79	69.13	66.05	3.08	0.04
10	0.87	136.15	132.05	4.10	0.03

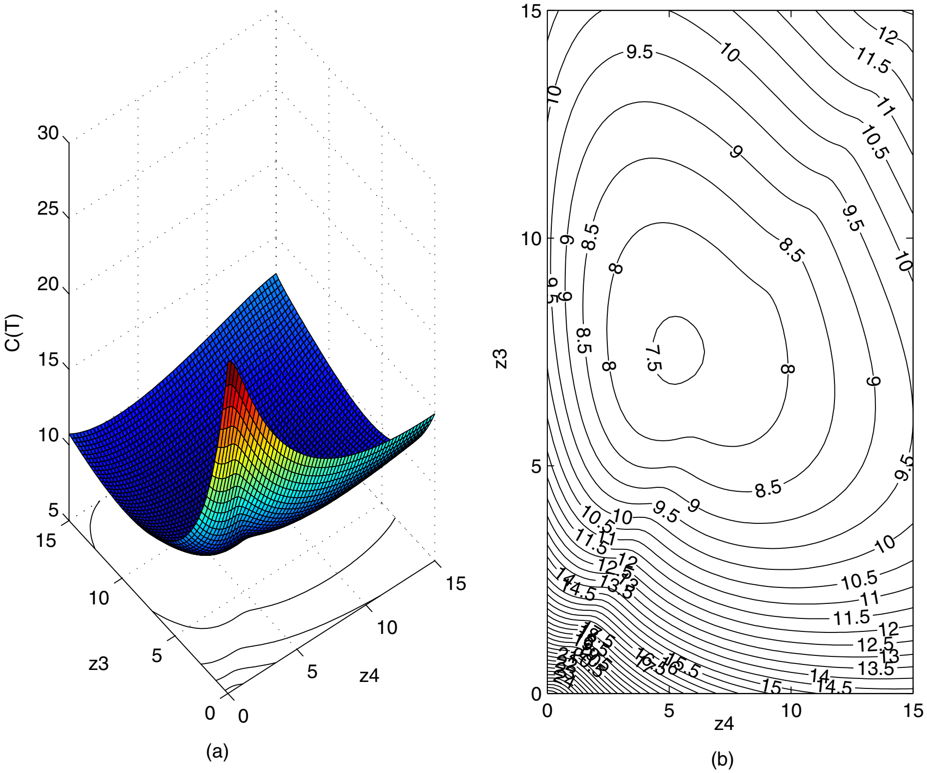

Fig. 5 Cost function of Example 1 with respect to hedging points of machine states 3 and 4

5.2.7 Clustering effect on hedging points

It can be observed from the optimal solutions of Examples 2 and 3 (Tables 6 and 8) that the optimal hedging points of some system states are very close to each other and form clusters. We have observed this behavior in several systems that we experimented with. Unfortunately, how to correctly predict which hedging points will form clusters is not obvious from the problem data in general. It mostly depends on the topology of the transition diagrams and the transition rates of machine and demand states.

5.3 Optimization of constrained production control problems

In this part, we report the results of two more numerical experiments. We modified our unconstrained minimization problem (3.1) to minimization of the probabilistically constrained problem (5.11) as given below:

$$\min_{z\geq 0} \lim_{T\to\infty} \frac{1}{T}\int_0^T c^+ X^+(t)dt$$

$$\text{s.t } P_{blog}(z) \leq \delta. \tag{5.11}$$

How to modify Eqs. 4.6 and 4.7 to account for the change of objective function is almost trivial and omitted. As mentioned earlier, the constraint function $P_{blog}(z)$ is the long run proportion of time that the system is in backlog state and it is desirable to keep it below a pre-specified constant δ. As T gets large, the long run proportion of time that the system is in backlog state can be computed as $P_{blog}(z) = B(T)/T$, where $B(T)$ is the cumulative time the system spends in backlog during T. Similar to the recursive expressions (4.6) and (4.7) respectively for $C(t_{n+1})$ and $dC(t_{n+1})/dz_{ij}$, it is possible to write recursive expressions for $B(t_{n+1})$ and $dB(t_{n+1})/dz_{ij}$ as follows:

$$\begin{aligned} B(t_{n+1}) &= B(t_n) + I_{\{X(t_n)<0\}} I_{\{X(t_{n+1})<0\}} \tau_{n+1} \\ &\quad + I_{\{X(t_n)\geq 0\}} I_{\{X(t_{n+1})<0\}} \frac{X(t_{n+1})}{u(t_n)} + I_{\{X(t_{n+1})\geq 0\}} I_{\{X(t_n)<0\}} \frac{-X(t_n)}{u(t_n)} \quad \text{and} \end{aligned}$$

$$\begin{aligned} \frac{dB(t_{n+1})}{dz_{ij}} &= \frac{dB(t_n)}{dz_{ij}} + I_{\{X(t_n)<0\}} I_{\{X(t_{n+1})<0\}} \frac{d\tau_{n+1}}{dz_{ij}} \\ &\quad + \frac{1}{u(t_n)} \left\{ I_{\{X(t_n)\geq 0\}} I_{\{X(t_{n+1})<0\}} \frac{dX(t_{n+1})}{dz_{ij}} - I_{\{X(t_{n+1})\geq 0\}} I_{\{X(t_n)<0\}} \frac{dX(t_n)}{dz_{ij}} \right\}. \end{aligned}$$

How these can be incorporated to the IPA algorithm in Appendix 4 is trivial and omitted.

Figure 6a shows the $P_{blog}(z)$ function of the system of Example 1 which is again non-convex, as one can clearly see from its level sets in Fig. 6b. Therefore, it is possible that when we start from different initial points, we may find different local optimal solutions which will indeed be the case for Examples 2′ and 3′ as explained below. Actually, we will explain that in Example 2′, this multiplicity is actually caused by multiple optima. Unfortunately, as in Example 3′, without additional testing, it is not generally possible to know the source of such multiplicity; i.e. whether it is due to several local isolated optima or multiple optima or both. Again, we also should note that despite the non-convexity of $P_{blog}(z)$, we did not encounter any numerical difficulties in our experiments in solving even larger problems.

Upon observing how flat both the objective function and the constraint around the solution points, we used 10^{-5} as $OptTol$ in Examples 2′ and 3′ below. In the presence of nonlinear constraints, one needs to specify an additional parameter, $ConsTol$, which is a measure of largest constraint violation that is acceptable at an optimal point; we set $ConsTol$ to 10^{-6}.

Example 2′ Here, we solved the probabilistically constrained problem for the system of Example 2 (of Section 5.1) with $\delta = 0.05$. In Table 11, it can be seen that when we start from different initial points (A and B), we find different optimal solutions (z^*_A and z^*_B, respectively). In Table 11, the more significant differences between the optimum hedging levels appears to be in z^*_{34} and z^*_{44}. In order to understand better the behavior of the objective function and the constraint, we draw both functions with respect to z_{34} and z_{44} by keeping the other hedging points at their optimal levels reached from the initial point B; see Figs. 7a and b. The contour lines of these functions illustrated in Fig. 7c show that the lowest contour line of the cost function lying below $P_{blog}(z) = 0.05$ is at 13.253 and it overlaps with the contour line of $P_{blog}(z) = 0.05$ at a range of hedging points. In this range, while z_{44} is anywhere above 27, z_{34} can vary between 30 and 32 and from the performance side, all these

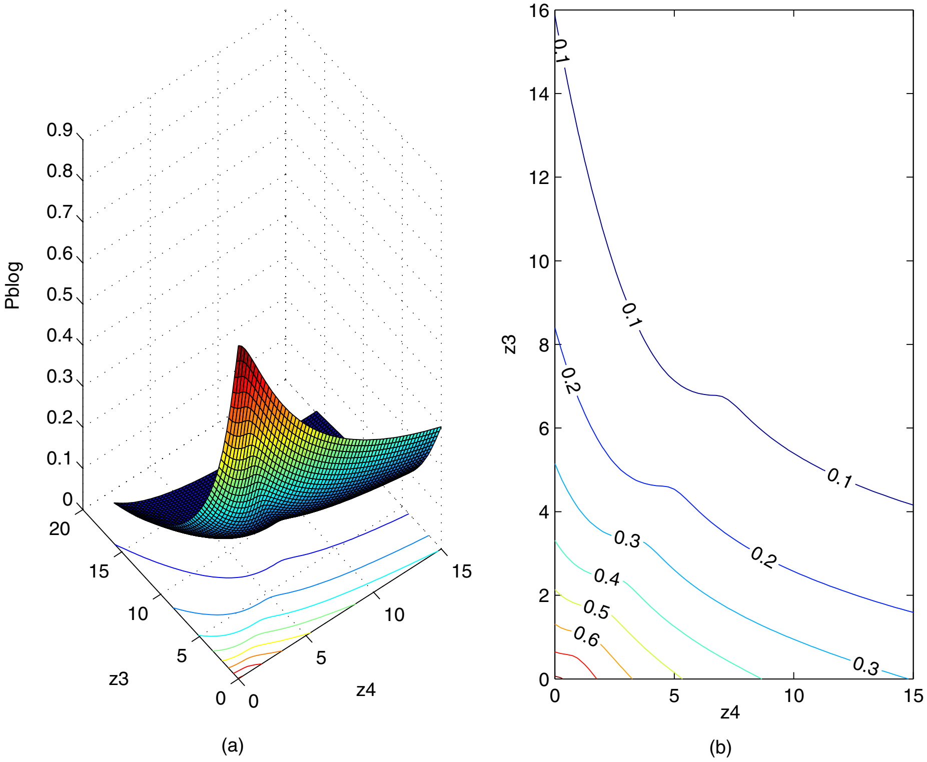

Fig. 6 $P_{blog}(z)$ of Example 1 with respect to the hedging points of machine states 3 and 4

points are equally good. We marked this range using a bold box in Fig. 7c. Indeed, both optimal solutions reported in Table 11 are located in that box, illustrating a case of multiple optima.

Example 3′*:* We also solved the probabilistically constrained problem for the system of Example 3 (of Section 5.1) with $\delta = 0.05$. The results are shown in Table 12.

Table 11 Optimal solution and the run-time information for Example 2′

Machine states	3	3	3	3	4	4	4	4	$J_T(z)$	$P_{blog}(z)$
Demand states	1	2	3	4	1	2	3	4		
Initial point *A*	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	2.812	0.261616
Initial point *B*	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00	20.596	0.036370
z_A^*	12.22	15.54	30.43	30.95	9.37	13.50	27.05	41.13	13.259	0.050000
z_B^*	12.22	15.62	30.30	31.41	9.41	13.53	27.29	28.03	13.253	0.050000

Initial point	N	Total CPU time
A	46	19 min 43 s
B	30	12 min 59 s

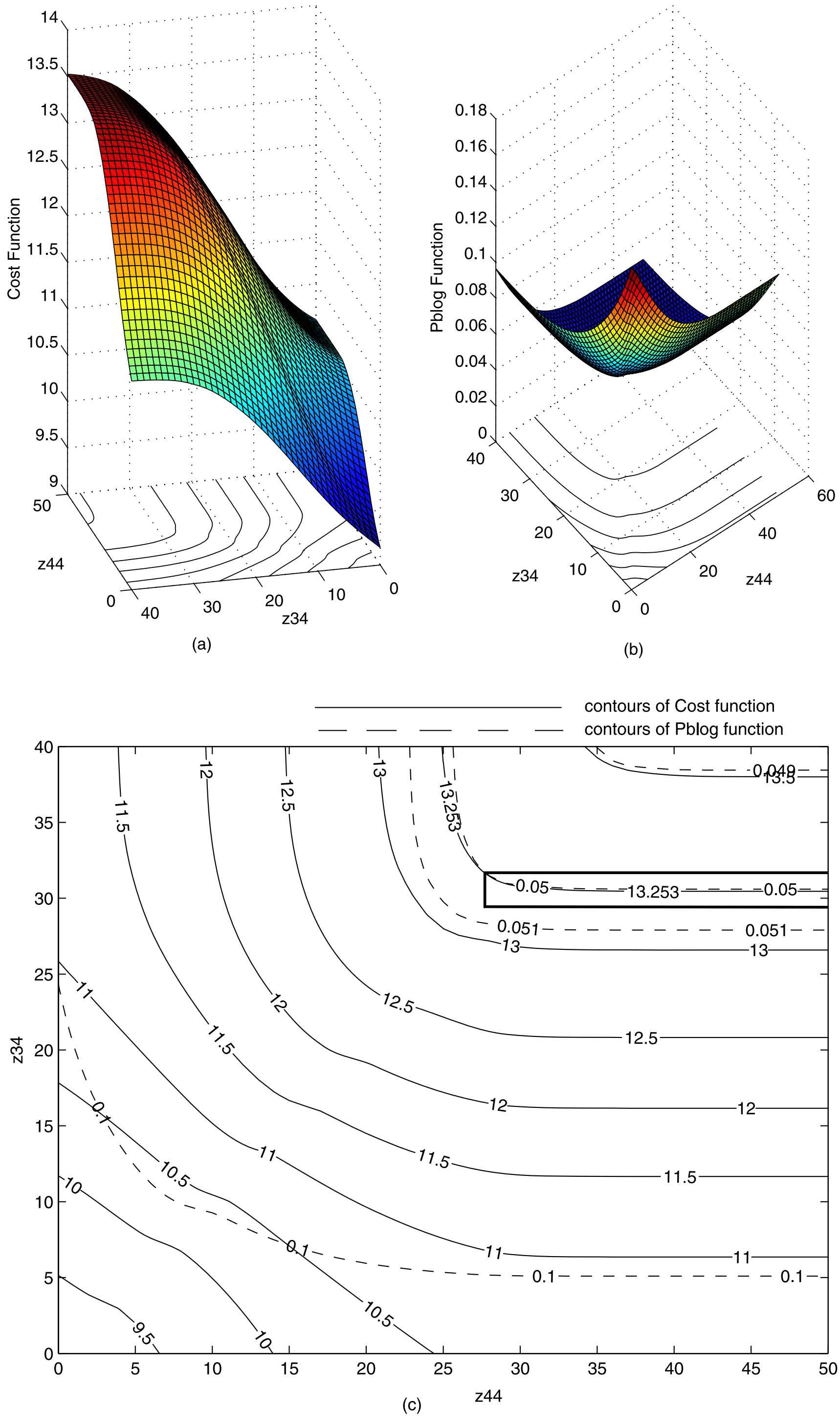

Fig. 7 Average cost and $P_{blog}(z)$ of Example 2′ with respect to z_{34} and z_{44}

Table 12 Optimal solution and the run-time information for Example 3′

Machine states	Demand states	Initial point *A*	Initial point *B*	Initial point *C*	z_A^*	z_B^*	z_C^*
4	1	5.00	10.00	20.00	57.92	25.71	26.88
4	2	5.00	10.00	20.00	26.43	28.56	24.82
4	3	5.00	10.00	20.00	64.33	49.75	40.39
4	4	5.00	10.00	20.00	82.52	52.16	37.49
5	1	5.00	10.00	20.00	25.85	28.26	28.12
5	2	5.00	10.00	20.00	26.65	34.22	28.56
5	3	5.00	10.00	20.00	69.72	48.23	39.02
5	4	5.00	10.00	20.00	75.45	52.24	38.01
7	1	5.00	10.00	20.00	14.32	15.22	14.97
7	2	5.00	10.00	20.00	16.97	16.78	16.59
7	3	5.00	10.00	20.00	45.99	42.67	30.05
7	4	5.00	10.00	20.00	40.13	39.54	32.70
8	1	5.00	10.00	20.00	15.25	14.64	14.48
8	2	5.00	10.00	20.00	17.38	17.98	17.44
8	3	5.00	10.00	20.00	42.78	29.03	29.57
8	4	5.00	10.00	20.00	58.12	39.23	29.48
9	1	5.00	10.00	20.00	15.25	15.69	15.61
9	2	5.00	10.00	20.00	17.56	16.31	20.75
10	1	5.00	10.00	20.00	21.78	22.56	21.32
10	2	5.00	10.00	20.00	21.99	14.53	23.02
10	3	5.00	10.00	20.00	65.56	45.69	37.79
10	4	5.00	10.00	20.00	75.00	42.19	35.19
$J_T(z)$		3.590	7.816	16.958	16.655	16.615	16.586
$P_{blog}(z)$		0.193072	0.123462	0.057995	0.050000	0.050000	0.050000

Initial point	N	Total CPU time
A	79	1 h 45 min 18 s
B	50	1 h 07 min 14 s
C	87	1 h 56 min 38 s

Similar to Example 2′, when we start from different initial points, we find different optimal solutions with very similar performances. In the absence of other constraints and/or other selection criteria, one can choose any one of these local optimal solutions since all perform equally well.

6 Conclusions

We investigated the problem of finding optimal hedging points in a stochastic fluid model of a production/inventory system by simulation-based optimization. The model is fairly general in terms of the number of machine and demand states and state transitions. On the modelling and simulation side, a GSMP representation provided a general and transparent description for the evolution of the system and was crucial for obtaining sample-path derivatives. On the optimization side, an implementation of sample-path optimization enabled us to optimize systems with more than twenty hedging points and probabilistic constraints.

The empirical performance of sample-path optimization in our numerical examples is very satisfactory. For small problems, the optimal solution always coincides with previously reported solutions. For large problems, although there are no reported benchmark solutions, the solutions obtained always pass a number of consistency checks. This is very promising for other potential applications of this approach and we consider it another encouraging sign that sample-path optimization will contribute to the solution of difficult problems with several variables and/or constraints.

The hedging point policy in the context of manufacturing flow control is an example of a general class of policies for stochastic fluid systems where the optimal control (i.e. controllable flow rate) for each state of the external stochastic process depends on a single threshold parameter expressed as the quantity of fluid in some buffer (inventory or queue). The flow rate is then adjusted depending on this threshold. A number of other interesting and important problems, other than manufacturing flow control, fall into this framework. In production/marketing problems involving dynamic pricing, the optimal price may vary as a function of thresholds (i.e. decrease price if inventory is higher than a threshold). In subcontracting/manufacturing problems, when to subcontract depends on the current inventory (or order queue) level (i.e. use subcontractor when the number of waiting orders exceeds a threshold). Similarly, in several admission control or routing problems in telecommunications, there is typically a threshold type policy structure (i.e. admit new arrivals of a given class if the buffer is lower than a threshold, reject otherwise). Our general approach should be applicable to such problems with certain modifications.

In addition, there are other important classes of stochastic optimization problems that are not typically modelled by stochastic fluid models but that involve the optimization of a finite number of threshold parameters in terms of certain buffer quantities. In revenue management which fare classes to open are typically determined by (possibly time-dependent) thresholds on the number of seats/resources available. Even though these problems may require different approaches for simulation and derivative estimation, the performance of sample-path optimization in our numerical results indicate that it is a promising approach for the optimization side of such problems.

Acknowledgement We thank Bariş Tan for fruitful discussions and useful comments on this problem.

Appendix 1: Proof of stability condition (3.2)

Recall that the state of the system at time t is $(\alpha(t), \beta(t))$ where $\alpha(t)$ is the state of the machine ($\alpha(t) = 1, 2, \ldots, K$) and $\beta(t)$ is the state of the demand ($\beta(t) = 1, 2, \ldots, M$), and the number of possible states of the system is KM, counting all combinations of $(\alpha(t), \beta(t))$. Tan (2002b) shows how a system with two machine states and two demand states can be represented by an equivalent system with four machine states coping with a constant demand rate. Below we generalize this basic idea for a system with KM states.

Consider a system state $k = (i, j)$ where $\alpha(t) = i, \beta(t) = j$. Let $\tilde{r}_k = r_i + \sum_{i=1}^{M} d_i - d_j$ be the maximum production rate and $p_k = \sum_{i=1}^{M} d_i - d_j$ be the minimum production rate associated with state k. Let $\tilde{d} = \sum_{i=1}^{M} d_i$ be the constant demand rate in the equivalent representation. Then we can label state k as deficient if $\tilde{r}_k < \tilde{d}$, non-deficient if $\tilde{r}_k > \tilde{d}$, and zero if $\tilde{r}_k = \tilde{d}$. This way state $k = (i, j)$ gets the same label in both representations:

(1) Deficient: if $\tilde{r}_k < \tilde{d}$ then $r_i + \sum_{i=1}^{M} d_i - d_j < \sum_{i=1}^{M} d_i$ and $r_i < d_j$,
(2) Non-deficient: if $\tilde{r}_k > \tilde{d}$ then $r_i + \sum_{i=1}^{M} d_i - d_j > \sum_{i=1}^{M} d_i$ and $r_i > d_j$,
(3) Zero: if $\tilde{r}_k = \tilde{d}$ then $r_i + \sum_{i=1}^{M} d_i - d_j = \sum_{i=1}^{M} d_i$ and $r_i = d_j$.

Let $\tilde{v}(t)$ denote the actual production rate and $\tilde{u}(t) = \tilde{v}(t) - \tilde{d}$ denote the fill-in rate at time t in the equivalent representation; fill-in rate specifies the total rate of change in the inventory level process at time t. The control $\tilde{v}(t)$ is again completely determined by the specifications of z_k's for non-deficient and zero states as:

$$\tilde{v}(t) = \begin{cases} \tilde{r}_k & \text{if } X(t) < z_k \\ \tilde{d} & \text{if } X(t) = z_k \\ p_k & \text{if } X(t) > z_k \end{cases}$$

and for deficient states $\tilde{v}(t) = \tilde{r}_k$ regardless of the inventory position.

Then for non-deficient and zero states $\tilde{u}(t)$ becomes:

$$\tilde{u}(t) = \begin{cases} \tilde{r}_k - \tilde{d} = r_i - d_j & \text{if } X(t) < z_k \\ \tilde{d} - \tilde{d} = 0 & \text{if } X(t) = z_k \\ p_k - \tilde{d} = -d_j & \text{if } X(t) > z_k \end{cases}$$

and for deficient states $\tilde{u}(t) = \tilde{r}_k - \tilde{d} = r_i - d_j < 0$ which again coincide with the fill-in rate, $u(t) = v(t) - d_{\beta(t)}$, of the original system representation.

Now, let ϕ_k $(k = 1, 2, .., KM)$ denote the stationary probability of being in system state $k = (i, j)$. Brémaud et al. (1997) show that a sufficient condition for stability of a system with KM machine states and a constant demand rate $\tilde{d}$ is given by:

$$\sum_{k=1}^{KM} \phi_k \tilde{r}_k > \tilde{d}. \tag{7.12}$$

The left side of Eq. 7.12 is $E[\tilde{r}]$, the expected maximum production rate of the system. $\tilde{r}_k$ is a random variable satisfying $\tilde{r}_k = r_i + \tilde{d} - d_j$. Hence, $E[\tilde{r}] = E[r] + \tilde{d} - E[d]$ with $E[r]$ being the expected maximum production rate of the machine and $E[d]$ being the expected demand. If we let π_i $(i = 1, 2, \ldots, K)$ to denote the stationary probability of being in machine state i and q_j $(j = 1, 2, \ldots, M)$ denote the stationary probability distribution of being in demand state j, then we can calculate $E[r]$ and $E[d]$ as follows: $E[r] = \sum_{i=1}^{K} \pi_i r_i$ and $E[d] = \sum_{j=1}^{M} q_j d_j$. After plugging these in the stability equation (7.12) and simplifying, we get:

$$\sum_{i=1}^{K} \pi_i r_i > \sum_{j=1}^{M} q_j d_j.$$

Appendix 2: Similarity properties

The sample path (obtained by fixing $\omega \in \Omega$) of the underlying stochastic process operated at the base hedging vector z is called the *nominal path*. The sample path (with the same ω) of the same process operated at a perturbed hedging vector $z + \Delta z_{ij}$ is called the *perturbed path*. Whenever we would like to indicate the value of a quantity in the perturbed path, we will put a superscript p to the corresponding quantity in the nominal path. For example, t_n^p denotes the occurrence time of nth event and $X^p(t)$ denotes the inventory level at time t, in the perturbed path.

Definition 1 A nominal path and a perturbed path are said to be *similar* in $[0, t_k]$ if and only if the order of events is the same for both paths; that is, the order of events in $[0, t_k]$ (in the nominal path) is the same as the order of events in $[0, t_k^p]$ (in the perturbed path).

As mentioned in Section 4, "similarity" of the nominal path and the perturbed path is a standard issue one needs to deal with when developing IPA algorithms and we address it in detail in this appendix. In particular, we show that along any sample path of finite length, say k events, with $min\ \{\tau_n | n = 1, 2, \ldots, k\} > 0$, there is always a $\Delta z_{ij} > 0$ (or $\Delta z_{ij} < 0$ whose size depends on the sample path) small enough such that increasing (or decreasing, respectively) z by Δz_{ij} does not cause any event order change; that is, the perturbed path and the nominal path remain similar.

To this end, notice that, under our "hit and stick" assumption, if t_n corresponds to the event $\mathcal{UH}$ or $\mathcal{DH}$, the length of time that $X(t)$ spends at z_{ij} (as a consequence of the nth event), τ_{n+1} satisfies: $\tau_{n+1} \geq \epsilon > 0$. In addition to"hit and stick", in the arguments below, without loss of generality we assume that either T (total simulation time) or the initial conditions are chosen in a way that ensures the end of simulation event $\mathcal{T}_f$ does not coincide with any other event.

Lemma 9 *Assume that the nominal and the perturbed paths (with an initial perturbation of Δz_{ij}) are similar in $[0, t_k]$. Then for all n $(n = 1, 2, ..k)$, the machine and demand states in the nominal path at the nth event time are equal to the machine and demand states in the perturbed path at the corresponding event time. That is $\alpha(t_n) = \alpha^p(t_n^p)$ and $\beta(t_n) = \beta^p(t_n^p)$ for all n $(n = 1, 2, ..k)$. Furthermore, if $e^*(t_n) \in \{\mathcal{MF}, \mathcal{DG}\}$, then $t_{n+1} = t_{n+1}^p$ and $e^*(t_n^p) = \mathcal{MF}$ (if $e^*(t_n) = \mathcal{MF}$) or $e^*(t_n^p) = \mathcal{DG}$ (if $e^*(t_n) = \mathcal{DG}$) for all n $(n = 1, 2, ..k)$.*

Proof The proof is by induction on the event instances t_n and is omitted. □

Lemma 10 *Assume that the nominal and the perturbed paths (with an initial perturbation of Δz_{ij}) are similar in $[0, t_k]$. Then $X^p(t) - X(t) \leq \Delta z_{ij}$, $\forall t \in [0, t_k]$. Furthermore, at all event times t_n $(n = 1, 2, .., k)$, we have $X^p(t_n^p) - X(t_n) = \Delta z_{ij}$ or $X^p(t_n^p) - X(t_n) = 0$.*

Proof We will only prove the second part of the lemma (the first part is an easy consequence) and will argue through induction. We start the induction at time $t_0 = 0$ with $\alpha(0) = i$, $\beta(0) = j$, and $X(0) = X^p(0) = x_0$. Without loss of generality, we assume that ij is a non-deficient system state. (Because in that case all possible events $\{\mathcal{MF}, \mathcal{UH}, \mathcal{DH}, \mathcal{DG}, \mathcal{T}_f\}$ may occur; whereas in zero or deficient states, only a subset of these events can take place.)

If $e^*(t_0) \in \{\mathcal{MF}, \mathcal{DG}, \mathcal{T}_f\}$, then $t_1 = t_1^p = \tau_1 = \tau_1^p$ and $X(t_1) = X^p(t_1^p)$. If $e^*(t_0) = \mathcal{UH}$, we should have $X(0) = X^p(0) = x_0 < z_{ij}$ and $u(t_1) = r_i - d_j$. With an initial perturbation of Δz_{ij}, we get $t_1^p = t_1 + \Delta z_{ij}/(r_i - d_j)$ and $X^p(t_1^p) = X(t_1) + \Delta z_{ij}$. Similarly, if $e^*(t_0) = \mathcal{DH}$, we should have $X(0) = X^p(0) = x_0 > z_{ij}$ and $u(t_1) = -d_j$. With Δz_{ij} as initial perturbation, we get $t_1^p = t_1 - \Delta z_{ij}/d_j$ and $X^p(t_1^p) = X(t_1) + \Delta z_{ij}$.

Now assume that, the stated property holds until (and including) the event transition times t_n and t_n^p. Let $\alpha(t_n) = k$ and $\beta(t_n) = m$. By Lemma 9, we have $\alpha^p(t_n^p) = k$ and $\beta^p(t_n^p) = m$. Next observe that we only need to consider the case where $k = i$ and $m = j$, because the others will leave everything unchanged.

We first consider the case $X(t_n) = X^p(t_n^p)$. If the last event at t_n is $\mathcal{MF}$ or $\mathcal{DG}$, that is $e^*(t_{n-1}) \in \{\mathcal{MF}, \mathcal{DG}\}$, then by Lemma 9, we have $t_n = t_n^p$. Hence, the induction step is the same as the beginning step. If $e^*(t_{n-1}) \in \{\mathcal{UH}, \mathcal{DH}\}$, as we are in state km with $k = i$ and $m = j$, the corresponding inventory levels at t_n and t_n^p should be $X(t_n) = z_{ij}$ and $X^p(t_n^p) = z_{ij} + \Delta z_{ij}$; this is a contradiction. Hence, we can conclude that if $X(t_n) = X^p(t_n^p)$, then $e^*(t_{n-1}) \in \{\mathcal{MF}, \mathcal{DG}\}$ and the induction step will be the same as the beginning step.

Next, we consider the case $X^p(t_n^p) = X(t_n) + \Delta z_{ij}$. We again need to differentiate two cases in which the last event can be either in $\{\mathcal{MF}, \mathcal{DG}\}$ or $\{\mathcal{UH}, \mathcal{DH}\}$.

If $e^*(t_{n-1}) \in \{\mathcal{MF}, \mathcal{DG}\}$, then by Lemma 9, we have $t_n = t_n^p$. Now we need to consider the possible events at t_{n+1}. If $e^*(t_n) \in \{\mathcal{MF}, \mathcal{DG}, \mathcal{T}_f\}$, then $\tau_{n+1} = \tau_{n+1}^p$. Since the system states are also the same, $X^p(t_n^p)$ and $X(t_n)$ will change equal amounts; that is $u(t_n) = u^p(t_n^p)$. Using $X(t_{n+1}) = X(t_n) + \tau_{n+1} u(t_n)$, we get $X^p(t_{n+1}^p) = X(t_{n+1}) + \Delta z_{ij}$. If $e^*(t_n) \in \{\mathcal{DH}, \mathcal{UH}\}$, as we are in state km with $k = i$ and $m = j$, the corresponding inventory levels at t_{n+1} and t_{n+1}^p should be $X(t_{n+1}) = z_{ij}$ and $X^p(t_{n+1}^p) = z_{ij} + \Delta z_{ij}$; hence the induction step holds.

If $e^*(t_{n-1}) \in \{\mathcal{DH}, \mathcal{UH}\}$, then the next event could be $\mathcal{MF}$, $\mathcal{DG}$, or $\mathcal{T}_f$; i.e. $e^*(t_n) \in \{\mathcal{MF}, \mathcal{DG}, \mathcal{T}_f\}$. Since the system will stay at its hedging level until the next event, and $X^p(t_n^p)$ and $X(t_n)$ will stay at the same levels with $u(t_n) = u^p(t_n^p) = 0$. Therefore, $X^p(t_{n+1}^p) = X(t_{n+1}) + \Delta z_{ij}$ and the induction step again holds. □

Remark 1 As a direct consequence of Lemma 10, we know that we can always select $\Delta z_{ij} > 0$ small enough such that if $X(t_n) \geq 0$ (or $X(t_n) < 0$), then $X^p(t_n^p) \geq 0$ (or $X^p(t_n^p) < 0$ respectively). This ensures that when $\Delta z_{ij} > 0$, $I_{\{X(t_n(z) \geq 0)\}} = I_{\{X(t_n(z+\Delta z y_{ij}) \geq 0)\}}$ and $I_{\{X(t_n(z) < 0)\}} = I_{\{X(t_n(z+\Delta z y_{ij}) < 0)\}}$ for any n.

Similarly as a consequence of Lemma 10, we can always select $\Delta z_{ij} < 0$ small enough such that if $X(t_n) > 0$ (or $X(t_n) < 0$), then $X^p(t_n^p) > 0$ (or $X^p(t_n^p) < 0$ respectively). When $X(t_n) = 0$, $e^*(t_{n-1})$ has to be either $\mathcal{DH}$ or $\mathcal{UH}$. In either case, we get $X^p(t_n^p) = 0$ as well. (To see $e^*(t_{n-1})$ cannot be $\mathcal{MF}$, $\mathcal{DG}$, or $\mathcal{T}_f$, realize that $X(t_n) = 0$ means $\tau_n = |X(t_{n-1}) u_{n-1}|$. This in turn would mean $|X(t_{n-1}) u_{n-1}| = \text{argmin}\,\{W^M(t_{n-1}),$ $W^D(t_{n-1}), T - t_{n-1}\}$ whose probability is zero since the probability of a continuous random variable being equal to a specific value is always zero).

Theorem 3 $\exists\, \epsilon_0 > 0$ *such that if any threshold* z_{ij} *is perturbed less than* ϵ_0 *(i.e.,* $|\Delta z_{ij}| < \epsilon_0$*), then the nominal and the perturbed paths remain similar.*

Proof Let $\Delta z_{ij} > 0$ be the initial perturbation. We will argue inductively as follows. Let $X(t_0) = X(0) = X^p(0)$ and let the initial machine state $\alpha(0) = k$ and $\beta(0) = m$ where without loss of generality, km is assumed to be a non-deficient state. First note that, if $km \neq ij$ then $e^*(t_0) = e^*(t_0^p)$, $t_1 = t_1^p$, and $X(t_1) = X^p(t_1^p)$ trivially. Therefore, we consider the case where the perturbation is associated to the hedging point

$km = ij$; that is $z^p_{km} = z_{km} + \Delta z_{km}$, with $\Delta z_{km} > 0$. By Lemma 9, we know that $e^*(t^p_0) = \mathcal{MF}$ if $e^*(t_0) = \mathcal{MF}$ or $e^*(t^p_0) = \mathcal{DG}$ if $e^*(t_0) = \mathcal{DG}$; hence $X(t_1) = X^p(t^p_1)$ and $t_1 = t^p_1$ trivially. The only interesting case happens if $e^*(t_0) \in \{\mathcal{UH}, \mathcal{DH}, \mathcal{T}_f\}$.

If $e^*(t_0) = \mathcal{UH}$, then this implies that $X(t_0) < z_{km}$ and $t_1 = \tau_1 = (z_{km} - X(0))/u(t_0)$. Now, the possible events in the perturbed path are $\{\mathcal{MF}, \mathcal{DG}, \mathcal{UH}$, and $T_f\}$. If $e^*(t^p_0) = \mathcal{MF}, \mathcal{DG}$, or T_f, then $t^p_1 = \tau^p_1 = W^M(t^p_0)$, $W^D(t^p_0)$, or T, respectively. If $e^*(t^p_0) = \mathcal{UH}$, then $t^p_1 = \tau^p_1 = t_1 + \Delta z_{km}/u(t_0)$ as illustrated in Fig. 8a. We need to show that if we select Δz_{km} small enough, an event other than $\mathcal{UH}$ does not occur on the perturbed path; that is $e^*(t^p_0) = \mathcal{UH}$. This can be guaranteed if $W^M(t^p_0) > t_1 + \Delta z_{km}/u(t_0)$, $W^D(t^p_0) > t_1 + \Delta z_{km}/u(t_0)$, and $T > t_1 + \Delta z_{km}/u(t_0)$. It is obvious that we can always ensure these conditions by selecting Δz_{km} such that $\Delta z_{km} < \min\{u(t_0)(W^M(t^p_0) - t_1), u(t_0)(W^D(t^p_0) - t_1), u(t_0)(T - t_1)\}$.

If $e^*(t_0) = \mathcal{DH}$, then this implies that $X(t_0) > z_{km}$ and $t_1 = \tau_1 = (X(0) - z_{km})/-u(t_0)$. Since $e^*(t_0) = \mathcal{DH}$, we know that $t_1 = \tau_1 < \min\{W^M(t_0), W^D(t_0), T\}$. The possible events for the perturbed path this time are $\{\mathcal{MF}, \mathcal{DG}, \mathcal{DH}$, and $T_f\}$ and $e^*(t^p_0) = \text{argmin}\,\{W^M(t^p_0), W^D(t^p_0), (X(0) - (z_{km} + \Delta z_{km}))/-u(t_0), T\}$. Since $W^M(t^p_0) = W^M(t_0)$, $W^M(t^p_0) = W^M(t_0)$, and $(X(0) - (z_{km} + \Delta z_{km}))/-u(t_0) < t_1$ from Fig. 8b, we get $e^*(t^p_0) = \mathcal{DH}$.

If $e^*(t_0) = T_f$ then $t_1 = \tau_1 = T$. We should consider two cases here: (a) In case $X(t_0) < z_{km}$, we know that $t_1 = T < \min\{W^M(t_0), W^D(t_0), (z_{km} - X(0))/u(t_0)\}$. The possible events for the perturbed path are $\{\mathcal{MF}, \mathcal{DG}, \mathcal{UH}$, and $T_f\}$ and $e^*(t^p_0) = \text{argmin}\,\{W^M(t^p_0), W^D(t^p_0), ((z_{km} + \Delta z_{km}) - X(0))/u(t_0), T\}$. Since $e^*(t_0) = T_f$, it is easy to see that $T < W^M(t^p_0) = W^M(t_0)$, $T < W^M(t^p_0) = W^M(t_0)$, and $T < ((z_{km} + \Delta z_{km}) - X(0))/u(t_0)$; therefore we get $e^*(t^p_0) = T_f$. (b) In case $X(t_0) > z_{km}$, we know that $t_1 = \tau_1 = T < \min\{W^M(t_0), W^D(t_0), (X(0) - z_{km})/-u(t_0)\}$. The possible events for the perturbed path are $\{\mathcal{MF}, \mathcal{DG}, \mathcal{DH}$, and $T_f\}$ and $e^*(t^p_0) = \text{argmin}\,\{W^M(t^p_0)$, $W^D(t^p_0), ((X(0) - (z_{km} + \Delta z_{km}))/-u(t_0), T\}$. We need to show that if we select Δz_{km} small enough, an event other than T_f does not occur on the perturbed path; that is $e^*(t^p_0) = T_f$. This can be guaranteed if $T < W^M(t^p_0)$, $T < W^D(t^p_0)$, and $T < ((X(0) - (z_{km} + \Delta z_{km}))/-u(t_0)$. We know that $T < W^M(t^p_0) = W^M(t_0)$, $T < W^M(t^p_0) = W^M(t_0)$ since $e^*(t_0) = T_f$. We can also ensure that $T < ((X(0) - (z_{km} + \Delta z_{km}))/-u(t_0)$

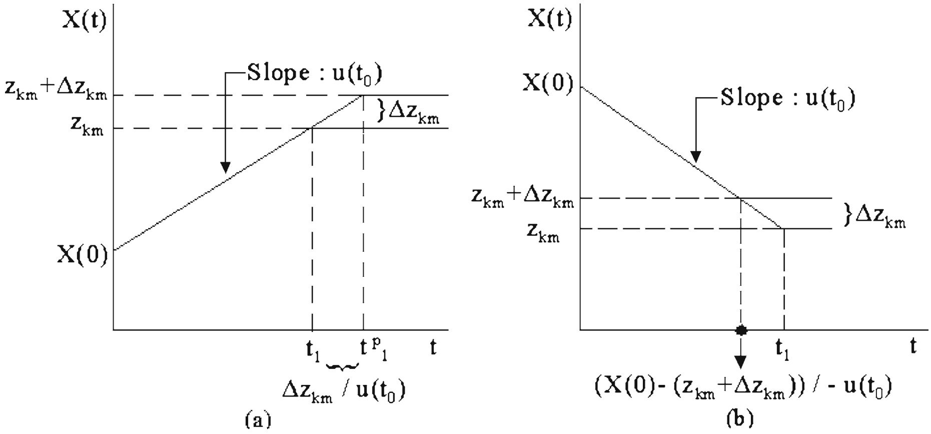

Fig. 8 Similarity of nominal and perturbed paths in $[0, t_1]$

holds by selecting Δz_{km} small enough $(0 < \Delta z_{km} < (X(0) - z_{km} + u(t_0)T))$. This proves the first step of the induction.

We now assume that the nominal path and the perturbed path are similar until (and including) the event times t_n and t_n^p. Let $\alpha(t_n) = k$ and $\beta(t_n) = m$. By Lemma 9, we have $\alpha^p(t_n^p) = k$ and $\beta^p(t_n^p) = m$. We have to show that the next events, on both nominal and the perturbed paths are identical; that is $e^*(t_n) = e^*(t_n^p)$.

By the argument in Lemma 10, we know that the similarity of the paths until the nth event implies that $X^p(t_n^p) = X(t_n)$ or $X^p(t_n^p) = X(t_n) + \Delta z_{ij}$. First assume that, $X^p(t_n^p) = X(t_n)$. If $ij = km$, then we know from the proof of Lemma 10 that $e^*(t_{n-1}) = e^*(t_{n-1}^p) \in \{\mathcal{MF}, \mathcal{DG}\}$ which implies that $t_n = t_n^p$. Also if $ij \neq km$ and if $e^*(t_{n-1}) = e^*(t_{n-1}^p) \in \{\mathcal{MF}, \mathcal{DG}\}$, then $t_n = t_n^p$. In both cases, $X^p(t_n^p) = X(t_n)$ and $t_n = t_n^p$; hence the rest of the argument is identical to the first step of the induction and the next events are identical. We now focus on the case where $ij \neq km$ but $e^*(t_{n-1}) = e^*(t_{n-1}^p) \in \{\mathcal{UH}, \mathcal{DH}\}$. This implies that $e^*(t_n)$ and $e^*(t_n^p)$ can only be a machine or demand state transition or end of simulation, on both nominal and perturbed paths; hence $e^*(t_n) = e^*(t_n^p)$.

Next assume that, $X^p(t_n^p) = X(t_n) + \Delta z_{ij}$. Here, we again need to consider two cases $ij = km$ and $ij \neq km$ separately.

In case $ij \neq km$, $e^*(t_{n-1}) = e^*(t_{n-1}^p)$ should be $\mathcal{MF}$ or $\mathcal{DG}$ (Because if $e^*(t_{n-1}) = e^*(t_{n-1}^p) \in \{\mathcal{UH}, \mathcal{DH}\}$ then $X(t_n) = X^p(t_n^p) = z_{km}$) and by Lemma 9 we have $t_n^p = t_n$. By Lemma 9, we again know that $e^*(t_n^p) = \mathcal{MF}$ if $e^*(t_n) = \mathcal{MF}$ or $e^*(t_n^p) = \mathcal{DG}$ if $e^*(t_n) = \mathcal{DG}$. If $e^*(t_n) = \mathcal{UH}$ (i.e. $X(t_n) < z_{km}$), then $\tau_{n+1} = (z_{km} - X(t_n))/u(t_n)$. The possible events for the perturbed path are $\{\mathcal{MF}, \mathcal{DG}, \mathcal{UH},$ and $T_f\}$ and we can always select Δz_{ij} small enough such that $X^p(t_n^p) = X(t_n) + \Delta z_{ij} < z_{km}$. Since $e^*(t_n) = \mathcal{UH}$, we know that $\tau_{n+1} < min\{W^M(t_n), W^D(t_n), T - t_n\}$. We also know that $e^*(t_n^p) = \text{argmin}\{W^M(t_n^p), W^D(t_n^p), (z_{km} - X^p(t_n^p))/u(t_n), T - t_n^p\}$. Since $W^M(t_n) = W^M(t_n^p)$, $W^D(t_n) = W^D(t_n^p)$, $T - t_n = T - t_n^p$ and $(z_{km} - X^p(t_n^p))/u(t_n) < \tau_{n+1}$ as illustrated in Fig. 9a, we get $e^*(t_n^p) = \mathcal{UH}$. If $e^*(t_n) = \mathcal{DH}$ (i.e. $X(t_n) > z_{km}$), then $\tau_{n+1} = (X(t_n) - z_{km})/ - u(t_n)$. The possible events for the perturbed path are $\{\mathcal{MF}, \mathcal{DG}, \mathcal{DH},$ and $T_f\}$ ($\mathcal{UH}$ is excluded since $X^p(t_n^p) > X(t_n) > z_{km}$). If $e^*(t_n^p) = \mathcal{MF}, \mathcal{DG}$, or T_f, then

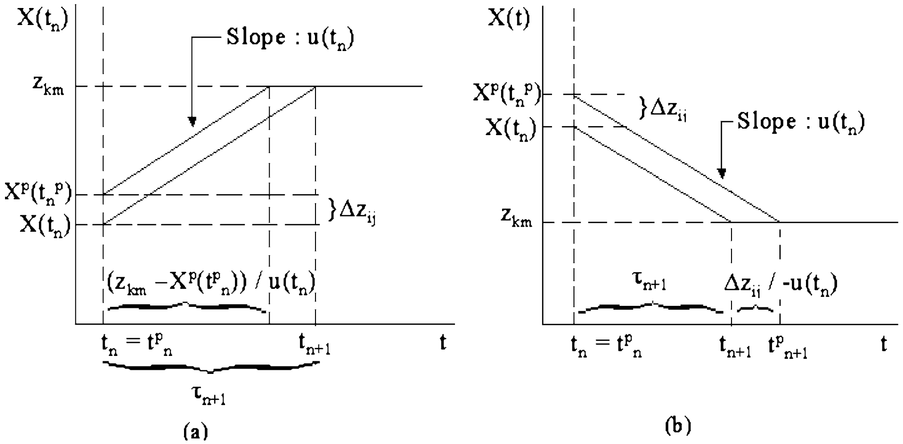

Fig. 9 Similarity of nominal and perturbed paths in $[t_n, t_{n+1}]$

$\tau^p_{n+1} = W^M(t^p_n)$, $W^D(t^p_n)$, or $T - t^p_n$, respectively. If $e^*(t^p_n) = \mathcal{DH}$ then $\tau^p_{n+1} = (X^p(t^p_n) - z_{km})/-u(t_n)$ which implies that $\tau^p_{n+1} = \tau_{n+1} + \Delta z_{ij}/-u(t_n)$ as illustrated in Fig. 9b. We need to show that if we select Δz_{ij} infinitesimally small enough, an event other than $\mathcal{DH}$ does not occur on the perturbed path, that is $e^*(t^p_n) = \mathcal{DH}$. This will be ensured if $W^M(t^p_n) > \tau_{n+1} + \Delta z_{ij}/-u(t_n)$, $W^M(t^p_n) > \tau_{n+1} + \Delta z_{ij}/-u(t_n)$, and $T - t^p_n > \tau_{n+1} + \Delta z_{ij}/-u(t_n)$. It is obvious that we can always ensure these conditions by selecting Δz_{ij} such that $\Delta z_{ij} < \min\{-(W^M(t^p_n) - \tau_{n+1})u(t_n), -(W^D(t^p_n) - \tau_{n+1})u(t_n), -(T - t^p_n - \tau_{n+1})u(t_n)\}$.

In case $ij = km$, we have $X^p(t^p_n) = X(t_n) + \Delta z_{km}$. First consider the case where $e^*(t_{n-1}) = e^*(t^p_{n-1}) \in \{\mathcal{MF}, \mathcal{DG}\}$; then we have $t^p_n = t_n$ by Lemma 9. By Lemma 9 again, $e^*(t^p_n) = \mathcal{MF}$ if $e^*(t_n) = \mathcal{MF}$ or $e^*(t^p_n) = \mathcal{DG}$ if $e^*(t_n) = \mathcal{DG}$. If $e^*(t_n) = \mathcal{UH}$ (i.e. $X(t_n) < z_{km}$), then $\tau_{n+1} = (z_{km} - X(t_n))/u(t_n)$. The possible events for the perturbed path are $\{\mathcal{MF}, \mathcal{DG}, \mathcal{UH}$, and $T_f\}(\mathcal{DH}$ is excluded since $X^p(t^p_n) = X(t_n) + \Delta z_{km} < z_{km} + \Delta z_{km})$. Since $e^*(t_n) = \mathcal{UH}$, we know that $\tau_{n+1} < min\{W^M(t_n), W^D(t_n), T - t_n\}$. We also know that $e^*(t^p_n) = \text{argmin}\{W^M(t^p_n), W^D(t^p_n), ((z_{km} + \Delta z_{km}) - X^p(t^p_n))/u(t_n), T - t^p_n\}$. Since $W^M(t_n) = W^M(t^p_n)$, $W^D(t_n) = W^D(t^p_n)$, $T - t_n = T - t^p_n$, and $((z_{km} + \Delta z_{km}) - X^p(t^p_n))/u(t_n) = \tau_{n+1}$, we get $e^*(t^p_n) = \mathcal{UH}$. The argument is similar for the case $e^*(t_n) = \mathcal{DH}$ with $\tau_{n+1} = (X^p(t^p_n) - (z_{km} + \Delta z_{km}))/-u(t_n)$. We now consider the case where $e^*(t_{n-1}) = e^*(t^p_{n-1}) \in \{\mathcal{UH}, \mathcal{DH}\}$. This implies that $e^*(t_n)$ and $e^*(t^p_n)$ can only be a machine or demand state transition or end of simulation, on both nominal and perturbed paths; hence $e^*(t_n) = e^*(t^p_n)$. This completes the induction argument.

The case with $\Delta z_{ij} < 0$ can be argued similarly. □

Appendix 3: Proofs of Lemmas in Section 4 for gradient estimation

Proof of Lemma 1 Observe that at any event time t_n, $r(e_{n+1}, t_n)$ is equal to either one of $0, 1, r_i - d_j$, or $-d_j$ for some i and j. Thus $r(e_{n+1}, t_n)$ is a function of r_i or d_j for some i and/or j and t_n. Of those variables only t_n depends on the hedging points. As a consequence of similarity for small enough Δz, $t_n(z + \Delta z y_{ij})$ is still the occurrence time of the nth event. Therefore we must have $r(e_{n+1}, t_n(z + \Delta z y_{ij})) = r(e_{n+1}, t_n(z))$ and

$$\frac{dr(e_{n+1}, t_n)}{dz_{ij}} = \lim_{\Delta z \to 0} \frac{r(e_{n+1}, t_n(z + \Delta z y_{ij})) - r(e_{n+1}, t_n(z))}{\Delta z} = 0. \qquad \square$$

Proof of Lemma 2 Similar to Lemma 1. □

Proof of Lemma 3 Since $u(t)$ is constant between adjacent events, we have $X(t_n) = X(t_{n-1}) + u(t_{n-1})\tau_n$. From Lemma 2, we know that $du(t_{n-1})/dz_{ij}$ is 0; hence the result follows. □

Proof of Lemma 4 Recall that $W^M(t_n) = W^M(t_{n-1}) - \tau_n$ and $W^D(t_n) = W^D(t_{n-1}) - \tau_n$. The result follows by taking the partial derivatives of these equations at z. □

Proof of Lemma 5 Recall that

$$k(e_{n+1}, t_n) = \begin{cases} W^M(t_n) & \text{if } e_{n+1} = \mathcal{MF}, \\ z_{\alpha(t_n)\beta(t_n)} - X(t_n) & \text{if } e_{n+1} = \mathcal{UH}, \\ X(t_n) - z_{\alpha(t_n)\beta(t_n)} & \text{if } e_{n+1} = \mathcal{DH}, \\ W^D(t_n) & \text{if } e_{n+1} = \mathcal{DG}, \\ T - t_n & \text{if } e_{n+1} = \mathcal{T}_f. \end{cases}$$

The result follows immediately. □

Proof of Lemma 6 Direct consequence of Eq. 4.5 and Lemma 1. □

Proof of Lemma 7 Since we start at time $t_0 = 0$, we have $X(t_0) = x_0$, $W^M(t_0) = W^M$, $W^D(t_0) = W^D$, and $C(t_0) = 0$. Clearly these are all independent of z_{ij}. Since $t_1 = \tau_1 = k(e_1, t_0)/r(e_1, t_0)$, it is enough to show the following facts about $k(e_1, t_0)$ and $r(e_1, t_0)$: (a) $r(e_1, t_0) \neq 0$; (b) $r(e_1, t_0)$ has partial derivatives at z for all z and $i = 1, \ldots, K$, $j = 1, \ldots, M$, which are finite; (c) $k(e_1, t_0)$ has partial derivatives at z for all z and $i = 1, \ldots, K$, $j = 1, \ldots, M$, which are finite. Since $e_1 \in E(t_0)$, (a) is immediate from the definition of $E(t)$ and (b) follows from Lemma 1. To see (c), observe that $E(t_0) = \{\mathcal{MF}, (\mathcal{UH} \text{ or } \mathcal{DH}), \mathcal{DG}, \mathcal{T}_f\}$. Hence we have

$$k(e_1, t_0) = \begin{cases} W^M & \text{if } e_1 = \mathcal{MF}, \\ z_{\alpha(t_0)\beta(t_0)} - x_0 & \text{if } e_1 = \mathcal{UH}, \\ x_0 - z_{\alpha(t_0)\beta(t_0)} & \text{if } e_1 = \mathcal{DH}, \\ W^D & \text{if } e_1 = \mathcal{DG}, \\ T - t_0 & \text{if } e_1 = \mathcal{T}_f. \end{cases}$$

Thus $k(e_1, t_0)$ is a differentiable function of z_{ij} for $i = 1, \ldots, K$, $j = 1, \ldots, M$ whose partial derivative is -1, 0, or 1. □

Proof of Lemma 8 From Lemma 10 in Appendix 2, we have $X(t_n(z + \Delta z_{ij}), \omega) - X(t_n(z), \omega) \leq \Delta z_{ij}$. Then $|C(t_n(z + \Delta z_{ij}), \omega) - C(t_n(z), \omega)| \leq t_n |\Delta z_{ij}|(c^+ + c^-)$. Hence $|C(T(z + \Delta z_{ij}), \omega) - C(T(z), \omega)| \leq T|\Delta z_{ij}|(c^+ + c^-)$ and $|J_T(z + \Delta z_{ij}, \omega) - J_T(z, \omega)| \leq |\Delta z_{ij}|(c^+ + c^-)$. Therefore

$$\left| \frac{J_T(z + \Delta z_{ij}, \omega) - J_T(z, \omega)}{\Delta z_{ij}} \right| \leq (c^+ + c^-).$$

□

Appendix 4: Pseudo code for simulation and IPA algorithm

4.1 Variables and procedures

Let w_i^M be the mean holding time in machine state i and w_j^D be the mean holding time in demand state j. We use sample(μ) to denote a sample taken from a prescribed distribution with mean μ.

The following events are defined for the GSMP representation when the current state is $\alpha\beta$:

$\mathcal{MF}$: The end of the sojourn time of the current machine state α
$\mathcal{UH}$: Reaching the hedging point $z_{\alpha\beta}$ of current system state $\alpha\beta$ from below

$\mathcal{DH}$: Reaching the hedging point $z_{\alpha\beta}$ of current system state $\alpha\beta$ from above
$\mathcal{DG}$: The end of the sojourn time of the current demand state β
$\mathcal{T}_f$: End of simulation

The following notation and procedures are used in the simulation/gradient estimation algorithm:

INF	: A very large constant
t	: Current time
τ	: Time to occurrence of triggering event
M	: Number of demand states
K	: Number of machine states
α	: Current machine state
β	: Current demand state
d_j	: Rate of the demand state j
z_{ij}	: Hedging point of state ij
r_i	: Maximum production rate of machine state i
v	: Current actual production rate of the system
u	: Current fill-in-rate of the system
X	: Current inventory level of the system
X_{prev}	: Previous inventory level of the system
$cost$	: Current cumulative cost of the system
Avg_cost	: Average cost of the system
No_Events	: Number of possible events
$E[i]$	: Possible event i, $i = 1, 2, \ldots, No\_Events$
$T[i]$	: Time to occurrence of $E[i]$
K^*	: Index of triggering event
W^M	: Time to occurrence of event $\mathcal{MF}$
W^D	: Time to occurrence of event $\mathcal{DG}$
T_{remain}	: Time to occurrence of event $\mathcal{T}_f$
t_{sim}	: Total simulation time
x_0	: Initial inventory level
α_0	: Initial machine state
β_0	: Initial demand state
$Csum[i][j]$	: Accumulator of dC/dz_{ij}
$Tsum[i][j]$	: Accumulator of dt_n/dz_{ij}
$Xsum[i][j]$	: Accumulator of $dX(t_n)/dz_{ij}$
$XPrevsum[i][j]$	: Accumulator of $dX(t_{n-1})/dz_{ij}$
$WMsum[i][j]$	: Accumulator of $dW^M(t_n)/dz_{ij}$
$WDsum[i][j]$	: Accumulator of $dW^D(t_n)/dz_{ij}$
$Temp[i][j]$	: Accumulator of $d\tau_n/dz_{ij}$
Gen_Sys	: Boolean variable which is 1 if it is a general system ($M > 1$), 0 otherwise
$S[i][j]$	: State of the system at machine state i and demand state j.

$$I_{\{\mathcal{A}\}} = \begin{cases} 1 \text{ if } \mathcal{A} \text{ holds} \\ 0 \text{ otherwise} \end{cases}$$

```
Procedure: Determine_Event_Space()
$E[0] = \mathcal{MF}$; $E[1] = \mathcal{UH}$; $E[2] = \mathcal{DH}$;
IF ($Gen\_Sys == 1$) THEN {$E[3] = \mathcal{DG}$; $E[4] = \mathcal{T}_f$; $No\_Events = 4$;}
ELSE {$E[3] = \mathcal{T}_f$; $No\_Events = 3$;}
```

```
Procedure: Determine_States()
FOR i=1 TO K DO
FOR j=1 TO M DO
    {IF ($r_i < d_j$) THEN $S[i][j] =$‘deficient’;
    ELSE IF ($r_i > d_j$) THEN $S[i][j] =$‘non-deficient’;
    ELSE IF ($r_i == d_j$) THEN $S[i][j] =$‘zero’;}
```

```
Procedure: Set_Uv()
IF ($S[\alpha][\beta] ==$‘non-deficient’) THEN
    IF ($X < z_{\alpha\beta}$) THEN $v = r_\alpha$;
    ELSE IF ($X == z_{\alpha\beta}$) THEN $v = d_\beta$;
    ELSE IF ($X > z_{\alpha\beta}$) THEN $v = 0$;
ELSE IF ($S[\alpha][\beta] ==$‘zero’) THEN
    IF ($X > z_{\alpha\beta}$) THEN $v = 0$; ELSE $v = r_\alpha$;
ELSE IF($S[\alpha][\beta] ==$‘deficient’) THEN $v = r_\alpha$;

$u = v - d_\beta$;
```

4.2 Simulation code

0. Initialization:

```
$t = 0$; $\tau = 0$; $X = x_0$; $X_{prev} = 0$; $T_{remain} = t_{sim}$; $cost = 0$;
IF ($M > 1$) THEN $Gen\_Sys = 1$; ELSE $Gen\_Sys = 0$ ;
Determine_Event_Space();
Determine_States();
Choose initial machine state $\alpha_0 : \alpha = \alpha_0$; Choose initial demand state $\beta_0 : \beta = \beta_0$;
Set_Uv();
Generate $W^M \sim$ sample($w_\alpha^M$)
IF ($Gen\_Sys == 1$) THEN Generate $W^D \sim$ sample($w_\beta^D$)
```

0(IPA). Initialization:

```
FOR i=1 TO K DO
FOR j=1 TO M DO
    {$Csum[i][j] = 0$; $Tsum[i][j] = 0$; $Xsum[i][j] = 0$; $XPrevsum[i][j] = 0$;
    $WMsum[i][j] = 0$;
    $Temp[i][j] = 0$;
    IF ($Gen\_Sys == 1$) THEN $WDsum[i][j] = 0$;}
```

1. Next Local Event:

```
$T[0] = W^M$;
IF ($S[\alpha][\beta]$ ==‘non-deficient’) THEN
    {IF ($X < z_{\alpha\beta}$) THEN { $T[1] = \frac{z_{\alpha\beta}-X}{u}$; $T[2] = INF$; }
    ELSE IF ($X == z_{\alpha\beta}$) THEN { $T[1] = INF$; $T[2] = INF$; }
    ELSE IF ($X > z_{\alpha\beta}$) THEN { $T[1] = INF$; $T[2] = \frac{X-z_{\alpha\beta}}{-u}$; }}
ELSE IF ($S[\alpha][\beta]$ ==‘zero’) THEN
    {$T[1] = INF$;
    IF ($X > z_{\alpha\beta}$) THEN $T[2] = \frac{X-z_{\alpha\beta}}{-u}$; ELSE $T[2] = INF$ }
ELSE IF($S[\alpha][\beta]$ ==‘deficient’) THEN
    {$T[1] = INF$; $T[2] = INF$}
IF ($Gen\_Sys == 1$) THEN {$T[3] = W^D$; $T[4] = T_{remain}$; } ELSE $T[3] = T_{remain}$;
```

2. Next Global Event:

```
$K^* =$argmin$(T[i] : i = 1, 2, \ldots, No\_Events)$;
$\tau = T[K^*]$; $e^* = E[K^*]$;
```

3A(IPA). Perturbation Generation:

```
Case:
$e^* = \mathcal{MF}$ : FOR i=1 TO K DO
                FOR j=1 TO M DO
                        $Temp[i][j] = WMsum[i][j]$;
$e^* = \mathcal{UH}$ :FOR i=1 TO K DO
                FOR j=1 TO M DO
                        $Temp[i][j] = \frac{1}{u}(I_{\{i=\alpha, j=\beta\}} - XPrevsum[i][j])$;
$e^* = \mathcal{DH}$ :FOR i=1 TO K DO
                FOR j=1 TO M DO
                        $Temp[i][j] = \frac{1}{-u}(-I_{\{i=\alpha, j=\beta\}} + XPrevsum[i][j])$;
$e^* = \mathcal{DG}$ :FOR i=1 TO K DO
                FOR j=1 TO M DO
                        $Temp[i][j] = WDsum[i][j]$;
$e^* = \mathcal{T}_f$ : FOR i=1 TO K DO
                FOR j=1 TO
                        $Temp[i][j] = -Tsum[i][j]$;
```

3B(IPA). Perturbation Update:

```
FOR i=1 TO K DO
FOR j=1 TO M DO
    {$Tsum[i][j] = Tsum[i][j] + Temp[i][j]$;
    $Xsum[i][j] = XPrevsum[i][j] + u \times Temp[i][j]$;
    IF $e^* = \mathcal{MF}$ THEN $WMsum[i][j] = 0$; ELSE $WMsum[i][j] =$
            $WMsum[i][j] - Temp[i][j]$;
    IF $e^* = \mathcal{DG}$ THEN $WDsum[i][j] = 0$; ELSE $WDsum[i][j] =$
            $WDsum[i][j] - Temp[i][j]$;}
4. Update:
$t = t + \tau$; $X_{prev} = X$; $X = X + u\tau$;

        $cost = cost + I_{\{X_{prev} \geq 0\}} I_{\{X \geq 0\}} c^+ \left[ X\tau - \frac{u\tau}{2}\tau \right]$
            $+ I_{\{X_{prev} < 0\}} I_{\{X < 0\}} c^- \left[ -X\tau + \frac{u\tau}{2}\tau \right]$
            $+ I_{\{X_{prev} \geq 0\}} I_{\{X < 0\}} \left\{ c^+ \left[ \frac{-X_{prev}}{2} \frac{X_{prev}}{u} \right] + c^- \left[ \frac{-X}{2} \frac{X}{u} \right] \right\}$
            $+ I_{\{X \geq 0\}} I_{\{X_{prev} < 0\}} \left\{ c^+ \left[ \frac{X}{2} \frac{X}{u} \right] + c^- \left[ \frac{X_{prev}}{2} \frac{X_{prev}}{u} \right] \right\}$

4A(IPA). Update Cost Perturbation:
FOR i=1 TO K DO
FOR j=1 TO M DO

{$Csum[i][j] = Csum[i][j]$
          $+ I_{\{X_{prev} \geq 0\}} I_{\{X \geq 0\}} c^+ [Xsum[i][j]\tau + XTemp[i][j] - u\tau Temp[i][j]]$
          $+ I_{\{X_{prev} < 0\}} I_{\{X < 0\}} c^- [-Xsum[i][j]\tau - XTemp[i][j] + u\tau Temp[i][j]]$
          $+ I_{\{X_{prev} \geq 0\}} I_{\{X < 0\}} \left\{ c^+ \left[ \frac{-X_{prev} XPrevsum[i][j]}{u} \right] + c^- \left[ \frac{-Xsum[i][j]X}{u} \right] \right\}$
          $+ I_{\{X \geq 0\}} I_{\{X_{prev} < 0\}} \left\{ c^+ \left[ \frac{Xsum[i][j]X}{u} \right] + c^- \left[ \frac{X_{prev} XPrevsum[i][j]}{u} \right] \right\}$

Case:
$e^* = \mathcal{MF}$: Generate new machine state $k$; $\alpha = k$; Generate $W^M \sim$ sample($w_\alpha^M$);
$e^* = \mathcal{DG}$: Generate new demand state $l$; $\beta = l$; Generate $W^D \sim$ sample($w_\beta^D$);

IF ($e^* \neq \mathcal{MF}$) THEN $W^M = W^M - \tau$;
IF ($Gen\_Sys == 1$) THEN
    IF ($e^* \neq \mathcal{DG}$) THEN $W^D = W^D - \tau$;
IF ($e^* = \mathcal{T}_f$) THEN GO TO Output;
ELSE IF ($e^* \neq \mathcal{T}_f$) THEN
    {$T_{remain} = T_{remain} - \tau$; Set_Uv(); GO TO Next Local Event;}

5. Output:
$Avg\_cost = cost/t_{sim}$;

5(IPA). Gradient Output:
FOR i=1 TO K DO
FOR j=1 TO M DO
    $dC/dz_{ij} = Csum[i][j]/t_{sim}$;
```

References

Akella R, Kumar PR (1986) Optimal control of production rate in a failure prone manufacturing system. IEEE Trans Automatic Control 31(2):116–126

Bielecki T, Kumar PR (1988) Optimality of zero-inventory policies for unreliable manufacturing systems. Oper Res 36(4):532–541

Birbil Şİ, Gürkan G, Listeş O (2006) Solving stochastic mathematical programs with complementarity constraints using simulation. Math Oper Res 31(4):739–760

Brémaud P, Malhamé RP, Massoulié L (1997) A manufacturing system with general stationary failure process: Stability and IPA of hedging control policies. IEEE Trans Automatic Control 42(2):155–170

Caramanis M, Liberopoulos G (1992) Perturbation analysis for the design of flexible manufacturing system flow controllers. Oper Res 40(6):1107–1125

Fletcher R (1987) Prac Methods Optim, 2nd edn. Wiley, Chichester

Fu MC, Healy K (1992) Simulation optimization of (s, S) inventory systems. In: Swain JJ, Goldsman D, Crain RC, Wilson JR (eds). Proceedings of the 1992 Winter Simulation Conference Piscataway, New Jersey: IEEE, pp 506–514

Gershwin SB (1994) Manufacturing system engineering. PTR Prentice Hall, Englewood Cliffs, New Jersey

Gill PE, Murray W, Wright MH (1981) Practical optimization. Academic Press, London

Glasserman P (1991) Gradient estimation via perturbation analysis. Kluwer, Norwell, MA

Glasserman P, Tayur S (1995) Sensitivity analysis for base-stock levels in multiechelon production-inventory systems. Manage Sci 41(2):263–281

Glynn PW (1989) A GSMP formalism for discrete event dynamic systems. Proc. of IEEE 77:14–23

Gürkan G (2000) Simulation optimization of buffer allocations in production lines with unreliable machines. Ann Oper Res 93:177–216

Gürkan G, Özge AY, Robinson SM (1996) Sample-path solution of stochastic variational inequalities, with applications to option pricing. In Charnes JM, Morrice DM, Brunner DT, Swain JJ (eds). Proceedings of the 1996 Winter Simulation Conference. IEEE, Piscataway, New Jersey, pp 337–344

Gürkan G, Özge AY, Robinson SM (1999a) Sample-path solution of stochastic variational inequalities. Math Program 84:313–333

Gürkan G, Özge AY, Robinson SM (1999b) Solving stochastic optimization problems with stochastic constraints: An application in network design. In: Farrington PA, Nembhard HB, Sturrock DT, Evans GW (eds). Proceedings of the 1999 Winter Simulation Conference. IEEE, Piscataway, New Jersey, pp 471–478

Haurie A, L'Ecuyer P, van Delft C (1994) Convergence of stochastic approximation coupled with perturbation analysis in a class of manufacturing flow control models. Discret Event Dyn Syst: Theory Appl 4:87–111

Ho Y-C, Cao X-R (1991) Perturbation analysis of discrete event dynamical systems. Kluwer, Norwell, MA

Hu J-Q, Xiang D (1995) Monotonicity of optimal control for failure-prone production systems. J Optim Theory Appl 86(1):57–71

Hu JQ, Vakili P, Huang L (2004) Capacity and production management in a single product manufacturing systems. Ann Oper Res 125:191–204

Kimemia J, Gershwin SB (1983) An algorithm for the computer control production in flexible manufacturing systems. IIE Trans 15:353–362

Kushner HJ, Clark DS (1978) Stochastic approximation methods for constrained and unconstrained systems. Springer-Verlag, New York, NY

L'Ecuyer P, Giroux N, Glynn PW (1994) Stochastic optimization by simulation: numerical experiments with the M/M/1 queue in the steady-state. Manage Sci 40:1245–1261

Leung YT (1990) Single-run optimization of discrete-event simulations. Ph.D. Dissertation, Department of Industrial Engineering, University of Wisconsin-Madison, Madison, USA

Liberopoulos G, Caramanis M (1994) Infinitesimal perturbation analysis for second derivative estimation and design of manufacturing flow controllers. J Optim Theory Appl 81(2):297–327

Liberopoulos G, Hu J-Q (1995) On the ordering of optimal hedging points in a class of manufacturing flow control models. IEEE Trans Automat Contr 40(2):282–286

Linderoth JT, Shapiro A, Wright SJ (2002) The empirical behavior of sampling methods for stochastic programming. Optimization Technical Report 02–01, University of Wisconsin-Madison, Madison, USA

Liu Y, Gong W (2002) Perturbation analysis for stochastic fluid queuing systems. Discret Event Dyn Syst: Theory Appl 12:391–416

Meketon M (1987) Optimization in simulation: A survey of recent results. In: Thesen A, Grant H, Kelton WD (eds). Proceedings of the 1987 Winter Simulation Conference. IEEE, Piscataway, New Jersey, pp. 58–67

Moré J, Wright SJ (1993) Optimization software guide. Frontiers in applied mathematics vol. 14. Society for industrial and applied mathematics, Philadelphia, Pennsylvania

NAG (2002) NAG C Library Manual, Mark 7. Oxford, England: NAG Ltd

Plambeck EL, Fu B-R, Robinson SM, Suri R (1993) Throughput optimization in tandem production lines via nonsmooth programming. In: Schoen J (ed). Proceedings of 1993 Summer Computer Simulation Conference. Society for Computer Simulation, San Diego, CA, pp 70–75

Plambeck EL, Fu B-R, Robinson SM, Suri R (1996) Sample-path optimization of convex stochastic performance functions. Math Program 75:137–176

Robbins H, Monro S (1951) A stochastic approximation method. Ann Math Stat 22:400–407

Robinson SM (1996) Analysis of sample-path optimization. Math Oper Res 21:513–528

Rubinstein RY, Shapiro A (1993) Discrete event systems: Sensitivity analysis and stochastic optimization by the score function method. Wiley, Chichester and New York

Sethi SP, Yan H, Zhang H, Zhang Q (2002) Optimal and hiearchical controls in dynamic stochastic manufacturing systems. Manuf Serv Oper Manag 4:133–170

Shapiro A, Homem-De-Mello T (1998) A simulation-based approach to two-stage stochastic programming with recourse. Math Program 81:301–325

Sharifnia A (1988) Production control of a manufacturing system with multiple machine states. IEEE Trans Automat Contr 33(7):620–625

Shedler GS (1993) Regenerative stochastic simulation. Academic Press, San Diego, CA

Sun G, Cassandras CG, Panayiotou CG (2004) Perturbation analysis of multiclass stochastic fluid models. Discret Event Dyn Syst: Theory Appl 14:267–307

Suri R, Fu B-R (1994) On using continuous flow lines to model discrete production lines. Discret Event Dyn Syst 4:129–169

Tan B (2001) Production control of a failure prone pull manufacturing system with variable demand. Working Paper No:2000/1 v2, Koç University, Istanbul, Turkey

Tan B (2002a) Managing manufacturing risks by using capacity options. J Oper Res Soc 53(2):232–242

Tan B (2002b) Production control of a pull system with production and demand uncertainty. IEEE Trans Automat Control 47(5):779–783

Tan B, Gershwin SB (2004) Production and subcontracting strategies for manufacturers with limited capacity and volatile demand. Ann Oper Res 125:205–232

Yan H, Yin G, Lou SXC (1994) Using stochastic optimization to determine threshold values for the control of unreliable manufacturing systems. J Optim Theory Appl 83(3):511–539

Yan H, Zhou XY, Yin G (1999) Approximating an optimal production policy in a continuous flow line: Recurrence and asymptotic properties. Oper Res 47(4):535–549

Yin G, Zhang Q, Yan HM, Boukas EK (2001) Random-direction optimization algorithms with applications to threshold controls. J Optim Theory Appl 110(1):211–233

Zhang Q, Yin GG, Boukas EK (2001) Optimal control of a marketing-production system. IEEE Trans Automat Contr 46(3):416–427

Zhao Y, Melamed B (2004) Make-to-Stock systems with backorders: IPA gradients. In: Ingalls R, Rosetti M, Smith J, Peters B (eds). Proceedings of the 2004 Winter Simulation Conference, IEEE, Piscataway, New Jersey, pp 559–567

Gül Gürkan is an Associate Professor of Operations Research and Management Science at the Department of Econometrics and Operations Research, within the Faculty of Economics and Business Administration at Tilburg University, in the Netherlands. She received her Ph.D. in Industrial Engineering from the University of Wisconsin–Madison in 1996. Her research interests include simulation, mathematical programming, stochastic optimization and equilibrium models, with applications in energy, economics, and production and service systems. She is a member of INFORMS and the Mathematical Programming Society.

Fikri Karaesmen is Associate Professor in the Department of Industrial Engineering at Koç University in Istanbul, Turkey. He has a Ph.D. from Northeastern University (Boston, MA). His research interests are in stochastic models of production and service systems. His papers have appeared in several journals including Operations Research, Management Science, IIE Transactions, and Manufacturing and Service Operations Management. He is on the editorial boards of IIE Transactions, IEEE Transactions on Automation Science and Engineering, and 4OR.

Özge Özdemir received her B.Sc. degree in Industrial Engineering from Boğaziçi University in İstanbul, Turkey in 2001 and her M.Sc. degree in Operations Research and Management from Tilburg University, in the Netherlands in 2003. Currently she is a Ph.D. student at the Department of Econometrics and Operations Research, Faculty of Economics and Business Administrations, Tilburg University. Her research interests are deterministic optimization, modeling and quantitative analysis of stochastic systems by using simulation-based optimization, with applications to management problems in several areas including energy planning, production-inventory systems, revenue management, and call centers.