Probabilistic Constrained Optimization: Methodology and Applications (S. P. Uryasev, Editor), pp. 45-66

On Optimization of Unreliable Material Flow Systems

Yu. Ermoliev (ermoliev@iiasa.ac.at)
International Institute for Applied Systems Analysis
A-2361 Laxenburg, Austria

S. Uryasev (uryasev@ise.ufl.edu)
University of Florida
474 Weil Hall, Gainesville, FL 32611-6595

J. Wessels (wessels@win.tue.nl)
Technical University Eindhoven
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

Abstract

The paper suggests an approach for optimizing a material flow system consisting of two work-stations and an intermediate buffer. The material flow system may be a production system, a distribution system or a pollutant-deposit/removal system. The important characteristics are that one of the work-stations is unreliable (random breakdown and repair times), and that the performance function is formulated in average terms. The performance function includes random production gains and losses as well as deterministic investment and maintenance costs. Although, on average, the performance function is smooth with respect to parameters, the sample performance function is discontinuous. The performance function is evaluated analytically under general assumptions on cost function and distributions. Gradients and stochastic estimates of the gradients were calculated using Analytical Perturbation Analysis. Optimization calculations are carried out for an example system.

1 Introduction

In several types of material flow systems, there is at least one unreliable component. This feature makes it particularly important to design such a system carefully by taking into account the uncertainties introduced by the component unreliability. Such material flow systems occur in production and distribution as well in environmental remediation (removal or transformation of pollutants) systems. In production systems, work-stations may be unreliable. In distribution systems, transport mechanisms may suffer from breakdowns. In environmental systems, the removal or transformation mechanism may be unavailable (due to climatic causes, for instance). Particular, for production systems, there is extensive literature on modeling and analysis of material flows with unreliable work-stations.

Analytic approaches have primarily been developed for the case of two work-stations with an intermediate buffer. For discrete products with deterministic processing times, we may refer to Buzacott [1] and Yeralan and Muth [25]. For continuous material flows with deterministic machine speeds, important references are Wijngaard [24] and Mitra [12]. De Koster [8] gives an overview of the literature and shows how to exploit Wijngaard's approach for the construction of a numerical procedure for the analysis of larger systems. Although these approaches are very valuable for getting a better understanding of the characteristics of relevant processes, they all suffer from the fact that they are based on severe assumptions. The usual requirement is that breakdown behavior as well as repair behavior is based on a negative-exponentially distributed time length or at least something very closely related to the negative-exponential distribution like a phase-type distribution with only a few phases. Therefore, for practical system design, simulation is a frequently used tool. However, a serious drawback of simulation is that a guided search for a good design usually requires many simulations. Particular, in the case of several design parameters, this can be prohibiting.

Various modeling and optimization approaches of the manufacturing systems are discussed in [11]. General approaches for optimizing stochastic systems by using Monte Carlo simulations follow from the the techniques of stochastic optimization (see, for example, [2, 10]). For Discrete Event Dynamic Systems, algorithms for evaluating unbiased estimates of the gradients are developed in [4, 7, 13, 14]. A disadvantage of the most of these algorithms is that they are not applicable when a sample-path is discontinuous in the relevant parameter, which is the case for number of material flow systems. Rubinstein [14] proposed a technique, which is called the Push Out method, to calculate sensitivities of systems with discontinuous sample-path. In [5], Gong and Ho suggested to smooth over the sample path function (Smoothed Perturbation Analysis) by taking conditional expectations w.r.t. a σ-algebra to estimate the gradient of the performance function. This approach in application to (s,S) inventory systems was explored in [16]. Similar ideas in combination with new differentiation formulas for probability functions [21, 22] were used in Analytic Perturbation Analy-

sis (APA) [23] to evaluate the performance function and the gradient during the same simulation run. In the framework of this approach, for systems with discontinuous sample-path, various estimates can be obtained, including the estimates of Smoothed Perturbation Analysis and the Push Out approach.

Paper [3] considered a system consisting of two machines (one is unreliable) and an intermediate buffer; based on the Monte-Carlo simulation optimization approach, it calculated gradient formulas for this system. In the present paper, for the same model, we show that using the ideas of Analytic Perturbation Analysis [23] the performance function and the gradients of this system can be calculated analytically for any distributions describing unreliability and repair characteristics of the machines. Although the sample path of this system is a discontinuous function, an average performance function is smooth with respect to control parameters. We found formulas for calculating the gradients of the performance function and their rough stochastic estimates. The paper discusses deterministic and stochastic optimization approaches for optimizing performance of the system. We evaluated with deterministic approach optimal parameters of an example system using the Variable Metric Algorithm [19] for nonsmooth optimization problems.

The following Section 2 introduces the model for a two-machine system with one unreliable machine and an intermediate buffer. Section 3 describes the optimization problem and provides analytical formulas for the discrete-continuous performance function and its gradients w.r.t. continuous variables. Section 4 considers two approaches for reducing discrete-continuous optimization problems to continuous ones. Deterministic and stochastic algorithms are described for the reduced problems. Stochastic estimates of the gradients are calculated with APA. Section 5 provides optimization results for an example system.

2 Description of the model

The system comprises two interacting processes (see Figure 1): a regular flow of material arriving at a "server" or "work-station" and a service process of this material. Each batch of material arrives at the server at equidistant points in time $t = 0,\ x_1,\ 2x_1, \ldots$ The intensity of this process can be adjusted by the value $x_1 > 0$. The work-station empties the available batches one-by-one and x_2 is the time needed to process a batch by the work-station. The processing of a batch can be interrupted by the failure of the work-station; therefore, the work-station goes through alternating "operation" and "repair" intervals. The lengths of these intervals are independent random variables with density functions ν and ϱ respectively. We suppose that a batch requires a position in the storage (buffer) from the time of its arrival until the moment that processing has been finished. Denote by S the number of batches that may be stocked in the storage. If the storage is full, we suppose that a newly arriving batch is lost with cost α. The gain of each processed batch is equal to β. The cost function also includes the investments and maintenance costs and provides a tradeoff

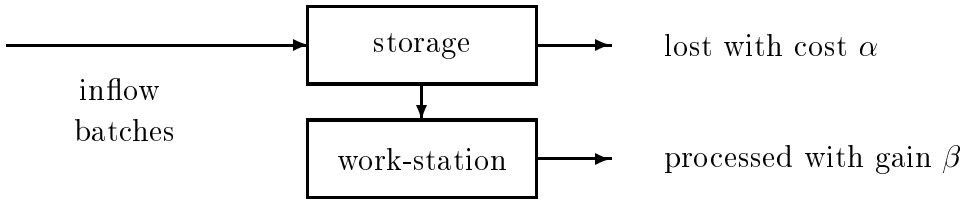

Figure 1: *System Flow Chart*

between profits and losses. In particular, in the case when $\beta \ll \alpha$, the main attention is paid to the losses due to exceeding storage capacity.

As a consequence of the interruptions, the real processing time of a batch may be essentially longer than x_2 . If T is the mean time to failure and R is the average repair time, then the availability fraction is

$$\frac{T}{R+T}$$

Consequently, the real processing time will be on the average

$$\frac{R+T}{T}\, x_2 \ .$$

If x_1 would be chosen smaller than this latter value, then the work-station could not cope with the input even if the storage capacity would have infinite size.

The model sketched above was inspired by the problem of designing a production system in which the batches would be delivered by a chain oven and the work-station treats the individual products of a batch one-by-one. In this way, the platter, which bears a batch of products through the oven, occupies a position in the buffer as long as it contains some products. It is not possible to stop the oven when the buffer is full, since this would lead to the loss of several hours of production; namely all the batches which are in the oven would be lost in this case. The chain pulls the batches through the oven with a fixed speed and batches which leave the oven are mechanically delivered to the buffer. If the buffer has no position available, then the batch is set aside and lost for further processing, since outside the buffer, the products cool down too much, which is not good for the quality.

In fact, the described system is an example of the two-machine system with one unreliable machine and an intermediate buffer. The problem is to design a system which works at minimal cost. There is no constraint on the output, because it is already certain that several units will be needed. Therefore, the only goal is to find the most efficient design. We denote by $c\,(x_1, x_2, S)$ the cost per unit of the processed product caused by investment and maintenance costs. We suppose that

this is a known function of the design parameters of the system. Also, there is a cost/gain component related to the performance of the system: each processed batch brings a gain β and lost batch a cost α. Therefore, the performance function equals

$$C(x,S) \;=\; c\,(x,S) \;+\; \alpha L(x,S) \;-\; \beta\Psi(x,S), \tag{1}$$

where $x \;=\; (x_1,x_2)\,,$ and $L(x,S)\,,\ \Psi(x,S)$ are the expected numbers of lost and processed batches per supplied batch, respectively.

Let us make some useful rearrangements of the problem. Denote by $\mathbf{L}_N(x,S)$ the random number of lost batches and by $\Psi_N(x,S)$ the random number of processed batches when N batches were supplied to the work-station. We use bold face style for random variables and functions. By definition

$$\mathbf{L}_N(x,S) \;+\; \Psi_N(x,S) \;=\; N\ ,$$

$$\lim_{N\to\infty}\ \frac{\mathbf{L}_N(x,S)}{N} \;=\; L(x,S)\,, \quad \lim_{N\to\infty}\ \frac{\Psi_N(x,S)}{N} \;=\; \Psi(x,S) \quad \text{(a.s.)}$$

Therefore, a sensible estimate of $\alpha L(x,S) - \beta\Psi(x,S)$ is

$$\alpha\,\frac{\mathbf{L}_N(x,S)}{N} \;-\; \beta\,\frac{\Psi_N(x,S)}{N} \;=\; \alpha\,\frac{\mathbf{L}_N(x,S)}{N} \;-\; \beta\,\frac{N-\mathbf{L}_N(x,S)}{N} \;=$$

$$(\alpha+\beta)\,\frac{\mathbf{L}_N(x,S)}{N} \;-\; \beta \;=\; \frac{(\alpha+\beta)\mathbf{L}_N(x,S)}{\mathbf{L}_N(x,S)+\Psi_N(x,S)} \;-\; \beta \;=$$

$$\frac{\alpha+\beta}{\mathbf{L}_N^{-1}(x,S)\Psi_N(x,S)+1} \;-\; \beta\ .$$

Let us denote

$$F(x,S) \;=\; \lim_{N\to\infty}\mathbf{L}_N(x,S)\Psi_N^{-1}(x,S) \quad \text{(a.s.)},$$

then

$$\alpha L(x,S) - \beta\Psi(x,S) \;=\; \frac{\alpha+\beta}{F^{-1}(x,S)+1} \;-\; \beta\ .$$

Thus the performance function equals

$$C(x,S) \;=\; c\,(x,S) \;+\; \frac{\alpha+\beta}{F^{-1}(x,S)+1} \;-\; \beta\,. \tag{2}$$

3 Optimization problem

Usually, feasible storage sizes $S \in \{S_1, \dots, S_I\}$ are known a priori. The problem is to find values x_1, x_2, and S_i such that the function $C(x, S_i)$ is minimal. Since $x_1 > x_2$, we included the following constraint

$$x_1 \geq x_2 + \kappa ,$$

where $\kappa > 0$. As we see further, the function $C(x, S)$ involves the calculation of expectations of discontinuous functions. Although, usually, an analytical evaluation of such functions is out of the question, for this particular case, using the ideas of Analytic Perturbation Analysis [23], we found an analytical expression for the function $C(x, S)$.

The performance function $C(x, S)$ can be calculated using the function $F(x, S)$. To evaluate the function $F(x, S)$, we supposed that the number of processed batches Ψ is fixed, so the number of batches $\mathbf{N}_\Psi(x, S)$ supplied to the work-station and the number of lost batches $\mathbf{L}_\Psi(x, S)$ are random functions of the variable Ψ and the control variables x, S. As an approximation of $F(x, S)$, we consider the function

$$F_\Psi(x, S) \;=\; \Psi^{-1}\, \mathbb{E}\, \mathbf{L}_\Psi(x, S) . \tag{3}$$

Although the function $\mathbf{L}_\Psi(x, S)$ is discontinuous w.r.t. x, the expectation of the function $\mathbf{L}_\Psi(x, S)$, which is an integral w.r.t. random variable, is a smooth function of x.

Let us denote:

ℓ is the number of an operation interval (an interval in which the work-station is available);

$\boldsymbol{\tau}_\ell$ is the number ℓ operation interval;

$\mathbf{r}_\ell$ is the repair interval after the operation interval $\boldsymbol{\tau}_\ell$;

$\mathcal{F}_\tau$ is the σ-algebra generated by the random operation intervals $\boldsymbol{\tau}_\ell$, $\ell = 1, 2, \dots$;

$\mathbb{E}_\tau$ is the conditional expectation w.r.t. the σ-algebra $\mathcal{F}_\tau$;

$\mathbb{P}_\gamma$ is the conditional probability w.r.t. the σ-algebra $\mathcal{F}_\gamma$.

We suppose that repair intervals, $\mathbf{r}_\ell$, $\ell = 1, 2, \dots$ and operation intervals, $\boldsymbol{\tau}_\ell$, $\ell = 1, 2, \dots$ are independent random values having densities, ϱ, and, ν, respectively. Also, to simplify calculations, we suppose that operation intervals cannot be shorter than some minimal value, $\boldsymbol{\tau}_{min}$, i.e., distribution function ν equals zero for the values less then $\boldsymbol{\tau}_{min}$. For example, ν could be a normal distribution truncated at the point $\boldsymbol{\tau}_{min}$. The value $\boldsymbol{\tau}_{min}$ is chosen such that the server, working during the time $\boldsymbol{\tau}_{min}$,

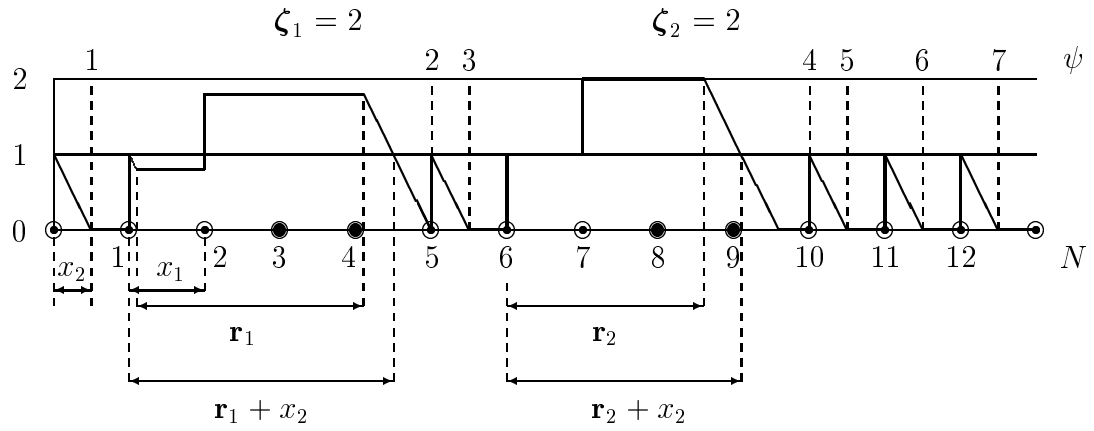

Figure 2: *Amount of material in the buffer. Buffer size equals 2. Batches with numbers 3, 4, 8, 9 are lost.*

is able to process the arriving batches and empty the buffer. Denote the maximal buffer size by

$$S_{max} = \max\{S_1, \dots, S_I\} \, .$$

If the server processes only the batches in the buffer, the maximal processing time of these batches is not longer than $x_2 S_{max}$. Since the server can devote only fraction $(x_1 - x_2)/x_1$ of its time to process the batches in the buffer, $\boldsymbol{\tau}_{min}$, must be larger than $x_2 S_{max}\, x_1/(x_1 - x_2)$.

Let us denote by $\mathbf{A}(x_2)$ a random number of failures of the work-station in a run consisting of Ψ processed batches. This number $\mathbf{A}(x_2)$ does not depend upon x_1, because the random operation interval $\boldsymbol{\tau}_\ell$ does not include idle time of the work-station.

For a large Ψ, the ratio of total processing time Ψx_2 and the mean time to failure T approximately equals the expected number of failures $\mathbb{E}\,\mathbf{A}(x_2)$, i.e.,

$$\mathbb{E}\,\mathbf{A}(x_2) \;=\; \Psi x_2/T \, . \tag{4}$$

Let us denote by $\boldsymbol{\zeta}_\ell(x, S)$ the number of lost batches because of the repair ℓ. The expectation of the function $\mathbf{L}_\Psi(x, S)$ can be represented as

$$\mathbb{E}\,\mathbf{L}_\Psi(x,S) \;=\; \mathbb{E}\sum_{\ell=1}^{\mathbf{A}(x_2)} \boldsymbol{\zeta}_\ell(x,S) \;=\; \mathbb{E}\mathbb{E}_\tau \sum_{\ell=1}^{\mathbf{A}(x_2)} \boldsymbol{\zeta}_\ell(x,S) \;=\; \mathbb{E}\sum_{\ell=1}^{\mathbf{A}(x_2)} \mathbb{E}_\tau\, \boldsymbol{\zeta}_\ell(x,S) \, , \tag{5}$$

Denote by $\mathbf{q}_\ell(x)$ the number of batches arrived to the work-station during repair ℓ plus the number of batches arrived during processing the remaining portion of the batch after finishing the repair ℓ. Thus, $\mathbf{q}_\ell(x)$ is the total number of batches

arrived to the work-station during processing the batch with repair ℓ. The number of lost batches, $\boldsymbol{\zeta}_\ell(x,S)$, can be expressed as the function of $\mathbf{q}_\ell(x)$ and S, i.e., $\boldsymbol{\zeta}_\ell(x,S) = K(\mathbf{q}_\ell(x),S)$. This number depends upon the size of the buffer: it equals zero, if $\mathbf{q}_\ell(x)$ is not larger than the size of the buffer, and equals $\mathbf{q}_\ell(x) - S$, if $\mathbf{q}_\ell(x)$ is larger than the buffer size S, i.e.,

$$K(q,S) \;=\; \max\{q-S,0\}\;. \tag{6}$$

With the full probability formula

$$G(x) \stackrel{\text{def}}{=} \mathbb{E}_\tau\,\boldsymbol{\zeta}_\ell(x,S) \;=\; \mathbb{E}_\tau K(\mathbf{q}_\ell(x),S) \;=\; \sum_{q=1}^{\infty} K(q,S)\phi_q(x)\;, \tag{7}$$

where

$$\phi_q(x) \stackrel{\text{def}}{=} \mathbb{P}_\tau\Big[\mathbf{q}_\ell(x) = q\Big]\;. \tag{8}$$

The processing time of the batch with repair ℓ, including repair time $\mathbf{r}_\ell$, equals $x_2 + \mathbf{r}_\ell$. The constraint $\mathbf{q}_\ell(x) = q$ is equivalent to the constraints (see Fig. 2).

$$q x_1 \;\le\; x_2 + \mathbf{r}_\ell \;\le\; (q+1)\,x_1\;. \tag{9}$$

Therefore, the function $\phi_q(x)$ can be calculated using a cumulative distribution function $D(r)$ of the random repair times $\mathbf{r}_\ell$

$$\phi_q(x) \;=\; D\Big((q+1)x_1 - x_2\Big) \;-\; D\Big(qx_1 - x_2\Big)\;. \tag{10}$$

Equations (4), (5), and (7) imply

$$\mathbb{E}\,\mathbf{L}_\Psi(x,S) \;=\; \mathbb{E}\,[\mathbf{A}(x_2)]\,G(x) \;=\; \frac{\Psi x_2}{T}\,G(x)\;.$$

Therefore, with (3) and (6)

$$F_\Psi(x,S) \;=\; T^{-1}x_2\,G(x) \;=\; T^{-1}x_2\sum_{q=S+1}^{\infty}(q-S)\,\phi_q(x)\;. \tag{11}$$

The function $\phi_q(x)$ tends to zero when q tends to infinity. Although formula (11) contains infinite summation, for practical purposes, it is sufficient to include only a finite number Q of terms. This number depends on the distribution function D and the values x_1 and x_2. Therefore, the function $F_\Psi(x,S)$ approximately equals

$$F(x,S) \;=\; T^{-1}x_2\sum_{q=S+1}^{Q}(q-S)\,\phi_q(x)$$

$$= \sum_{q=S+1}^{Q} (q-S)\left[D\big((q+1)x_1 - x_2\big) \;-\; D\big(qx_1 - x_2\big)\right]$$

$$= \sum_{q=S+1}^{Q} \left[(q-S)\,D\big((q+1)x_1 - x_2\big) \;-\; (q-1-S)\,D\big(qx_1 - x_2\big) \;-\; D\big(qx_1 - x_2\big)\right]$$

$$= T^{-1}x_2 \left((Q-S)D\big((Q+1)x_1 - x_2\big) - \sum_{q=S+1}^{Q} D\big(qx_1 - x_2\big)\right) . \qquad (12)$$

Thus, finally, we came to the following minimization problem:

Minimization problem

$$C(x,S) \;\rightarrow \min_{\substack{x\in X\,,\\ S\in\{S_1,\dots,S_I\}}} \qquad (13)$$

subject to

$$X \;=\; \{\, x\in\mathbb{R}^2 \;:\; x_1 \;\geq\; x_2+\kappa\,,\; x_2 \;\geq\; 0\,\}\,, \qquad (14)$$

where the performance function equals

$$C(x,S) \;=\; c\,(x,S) \;+\; \frac{\alpha+\beta}{F^{-1}(x,S)+1} \;-\; \beta\,, \qquad (15)$$

and the function $F(x,S)$ is given by equation (12).

4 Approaches to solve the optimization problem

This section discusses optimization approaches for solving the mixed discrete-continuous optimization problem (13) with constraints (14). Further, we consider two approaches for reducing this problem to a continuous one. Also, we described deterministic and stochastic algorithms for solving this reduced continuous optimization problem.

4.1 Decomposition approach

The optimization problem (13) is continuous w.r.t. x and is discrete w.r.t. S. If the buffer size S_i is fixed, this is a typical nonlinear optimization problem with linear constraints with respect to x. For each buffer size S_i, we can solve the problem

$$C(x,S_i) \;\rightarrow \min_{x\in X} \qquad (16)$$

with respect to x and find an optimum vector x_i^*. Then, we can find optimal buffer size S_i minimizing $C(x_i^*, S_i)$ w.r.t. S_i. Because for each buffer size we need to run a nonlinear programming algorithm, this brute-force approach is applicable for the problems with a relatively small number of feasible buffer sizes S_i. The function $C(x, S)$ is smooth w.r.t. the variable x for the fixed buffer size S; moreover the gradient w.r.t. x can be calculated in analytical form. Indeed, we supposed that the first term $c(x, S)$ of the performance function $C(x, S)$ is smooth; also, the second term is smooth because it is expressed (see (12)) through the smooth function $F(x, S)$. Since the function $D\big(qx_1 - x_2\big)$ can be analytically differentiated w.r.t. x_1 and x_2, i.e.,

$$\frac{\partial}{\partial x_1} D\big(qx_1 - x_2\big) \;=\; q\,\varrho\,\big(qx_1 - x_2\big)\;, \quad \frac{\partial}{\partial x_2} D\big(qx_1 - x_2\big) \;=\; -\varrho\,\big(qx_1 - x_2\big)\;, \tag{17}$$

the function $F(x, S)$ and, consequently, the performance function $C(x, S)$ can be analytically differentiated w.r.t. x_1 and x_2. So, we can use efficient nonlinear gradient algorithms to solve subproblem (16).

4.2 Artificial variables approach

An alternative way to reduce problem (13) to a continuous optimization problem is using artificial variables. It can be proved that problem (13) is equivalent to the minimization problem

$$\Phi(x, y) \;\stackrel{\text{def}}{=}\; \sum_{i=1}^{I} C(x, S_i)\, y_i \;\;\to \min_{(x,y)\in \mathbb{R}^2\times\mathbb{R}^I}\;, \tag{18}$$

subject to constraints

$$\sum_{i=1}^{I} y_i \;=\; 1\;; \quad y_i \ge 0\,,\; i = 1, \ldots, I\;; \tag{19}$$

$$x \in X\;. \tag{20}$$

Denoting

$$Y \;=\; \{\, y \in \mathbb{R}^I \;:\; \sum_{i=1}^{I} y_i \;=\; 1\,,\; y_i \ge 0\,,\; i = 1, \ldots, I \,\}\;,$$

the problem (13) can be reformulated as

$$\Phi(x, y) \;\to \min_{(x,y)\in X\times Y}\;. \tag{21}$$

This is a continuous optimization problem with linear constraints. Despite the fact that the original problem is a mixed discrete-continuous optimization problem, we reduced it to a problem with continuous variables by using additional variables $y_1, \ldots, y_I$. To calculate the performance function $\Phi(x, y)$, the function $C(x, S)$ should be calculated I times. A potential problem with this approach is that for the case with large I , the exact evaluation of the function $\Phi(x, y)$ may involve a tremendous amount of calculations. The next section shows that these numerical difficulties can be overcome with stochastic quasigradient algorithms, which use only rough stochastic estimates of the gradients of the performance function.

4.3 Stochastic quasigradient algorithm

As we mentioned in the previous section, for problem (21) nonlinear programming methods may be a poor choice, because of prohibitively large amount of calculations involved in evaluating the performance function and the gradients. This section considers an alternative approach which is called *stochastic quasigradient* algorithms (on background of stochastic quasigradient algorithms see [2]). One of the most simple stochastic quasi-gradient algorithms for problem (21) can be represented in the following form

$$(x^{s+1}, y^{s+1}) = \Pi_{X \times Y}\Big((x^s, y^s) - \rho_s \xi^s\Big) \ , \tag{22}$$

where s is a number of algorithm iterations; (x^s, y^s) is the approximation point of the extremum on the s^{th} iteration; $\Pi_{X \times Y}(\cdot)$ is the orthoprojection operation on the convex set $X \times Y$; $\rho_s > 0$ is a step size; and ξ^s is a stochastic quasi-gradient satisfying the following property

$$E\,[\,\xi^s \mid (x^0, y^0), \ldots, (x^s, y^s)\,] = \nabla_x \Phi(x^s, y^s) \ .$$

The conditional expectation of the vector ξ^s is equal to the gradient of the function $\Phi(x, y)$ at the point (x^s, y^s). This algorithm is quite efficient for non ill-conditioned performance functions, i.e., for non-"ravine" functions. In case when the function $\Phi(x, y)$ is "ravine", algorithm (22) may get stuck "at the bottom of the ravine". In such a case, more complicated stochastic quasigradient algorithms, such as algorithms with averaging (see, for example [6], [9],[17]) or variable metrics algorithm [20] may be used. Also, practical convergence rate of algorithm (22) may be improved using adaptively controlled step sizes [18] and the scaling procedure, suggested by Saridis [15].

Further, we provide formulas for calculating stochastic quasigradients of the function $\Phi(x, y)$. An advantage of the stochastic quasigradient algorithms is that they may use rough stochastic estimates of the gradient, which can be obtained with very little computational effort compared to the effort needed for calculating the exact gradient of the performance function $\Phi(x, y)$. Formula (15) implies that

$$C(x, S) \ = \ \Lambda\Big(x, S, F(x, S)\Big)$$

and

$$\Phi(x,y) \;=\; \sum_{i=1}^{I} \Lambda\Big(x, S_i, F(x,S_i)\Big)\, y_i \tag{23}$$

Hence,

$$\nabla_x \Phi(x,y) \;=\; \sum_{i=1}^{I} y_i \,\big[\nabla_x \Lambda(x,S_i,z) + \nabla_z \Lambda(x,S_i,z) \nabla_x F(x,S_i)\big]_{z=F(x,S_i)} \;, \tag{24}$$

$$\nabla_y \Phi(x,y) \;=\; \left(\begin{array}{c} \Lambda\Big(x, S_1, F(x,S_1)\Big) \\ \vdots \\ \Lambda\Big(x, S_I, F(x,S_I)\Big) \end{array} \right) . \tag{25}$$

To calculate stochastic quasigradients, instead of exact values $F(x,S_i)\,,\ \nabla_x F(x,S_i)\,,$ $i = 1,\ldots,I$ in formulas (24) and (25), we can use rough stochastic estimates. In this case, estimates of the gradients $\nabla_x \Phi(x,y)\,,\ \nabla_y \Phi(x,y)$ are biased, because the function $\Lambda\Big(x,S_i,z)\Big)$ is nonlinear w.r.t. z. However, this bias is relatively small because the function $\Lambda\Big(x,S_i,z)\Big)$ is close to linear w.r.t. z. Indeed, by definition, the function $F(x,S)$ is a ratio of the number of lost and processed batches. From engineering considerations, this number is much less than 1. Therefore, the function $\Lambda\Big(x,S_i,F(x,S)\Big)$ approximately equals the following linear function $F(x,S)$

$$\Lambda\Big(x,S,F(x,S)\Big) \;=\; C(x,S) \;=\; c\,(x,S) \;+\; \frac{\alpha+\beta}{F^{-1}(x,S)+1} \;-\; \beta$$

$$\approx\; c\,(x,S) \;+\; (\alpha+\beta) F(x,S) \;-\; \beta\,.$$

As follows from (7) and (11) the function $F(x,S)$ equals

$$F_\Psi(x,S) \;=\; T^{-1} x_2\, G(x) \;=\; T^{-1} x_2\, \mathbb{E}_\tau\, \boldsymbol{\zeta}_\ell(x,S)\,. \tag{26}$$

Consequently, $T^{-1} x_2\, \boldsymbol{\zeta}_\ell(x,S)\,,$ is an unbiased estimate of the function $F(x,S)$ which can be obtained by sampling the lost number of batches $\boldsymbol{\zeta}_\ell(x,S)\,.$ Formula (26) implies that the gradient of the function $F_\Psi(x,S)$ equals

$$\nabla_x F_\Psi(x,S) \;=\; T^{-1} \left(\begin{array}{c} 0 \\ 1 \end{array} \right) \;+\; T^{-1} x_2\, \nabla_x G(x)\,. \tag{27}$$

Gradient of the function $G(x)$ can be calculated using APA [23]; see brief description of APA in Appendix A. Similar to (35), equation (7) represents the function $G(x)$ as a sum of indicator functions, where (see (8),(9))

$$\phi_q(x) \;=\; \mathbb{E}_\tau\Big[I_{\{q x_1 \,\le\, x_2 + \mathbf{r}_\ell \,\le\, (q+1)\, x_1\}}\Big] \;=\; \int\limits_{q x_1 \,\le\, x_2 + r \,\le\, (q+1)\, x_1} \varrho(r)\; \mathrm{d}r$$

$$= \int\limits_{q \le (x_2+r)/x_1 \le q+1} \varrho(r)\, \mathrm{d}r \ . \tag{28}$$

To use APA, the gradient $\nabla_x \phi_q(x)$ should be represented in a form similar to the integral (28). Since analytical expression (10) for $\phi_q(x)$ and its derivative (see (17)) is available, we can write

$$\nabla_x \phi_q(x) \ = \ \frac{\nabla_x \phi_q(x)}{\phi_q(x)}\ \phi_q(x) \ = b^q(x)\, \phi_q(x) \ ,$$

where

$$b^q(x) \ = \ \frac{\left(\begin{array}{c} (q+1)\varrho\Big((q+1)x_1 - x_2\Big) \ - q\ \varrho\Big(qx_1 - x_2\Big) \\ -\varrho\Big((q+1)x_1 - x_2\Big) \ + \ \varrho\Big(qx_1 - x_2\Big) \end{array} \right)}{\Big(D\Big((q+1)x_1 - x_2\Big) \ - \ D\Big(qx_1 - x_2\Big)\Big)} \ .$$

As it follows from formula (39) in Appendix A, a unbiased estimate of the gradient $\nabla_x G(x)$ equals

$$K(\mathbf{q}_\ell(x), S)\, b^{\mathbf{q}_\ell}(x) \, , \tag{29}$$

and

$$\nabla_x G(x) \ = \ \mathbb{E}_\tau[K(\mathbf{q}_\ell(x), S)\, b^{\mathbf{q}_\ell}(x)] \ . \tag{30}$$

Also, this formula can be obtained with Smoothed Infinitesimal Perturbation Analysis [5]. APA provides one more expression for the estimate of the gradient of the function $G(x)$, which is based on integral over volume formula (44). Since the change of variables $z = (x_2 + r)/x_1$ eliminates variables x_1, x_2 from constraints in integral (28), the matrix $H(x, r)$ can be calculated with formula (46)

$$H(x,r) \ = \ \nabla_x(x_1 z \ - \ x_2)|_{z=(x_2+r)/x_1} \ = \ \left(\begin{array}{c} z \\ -1 \end{array} \right)\Bigg|_{z=(x_2+r)/x_1} = \ \left(\begin{array}{c} (x_2+r)/x_1 \\ -1 \end{array} \right) \ .$$

Therefore, with formula (44), the gradient of integral (28) equals

$$\nabla_x \phi_q(x) \ = \int\limits_{q \le (x_2+r)/x_1 \le q+1} \frac{\partial}{\partial r}\Big(\varrho(r)\, H(x,r)\Big)\, \mathrm{d}r \ = \int\limits_{q \le (x_2+r)/x_1 \le q+1} a^q(x)\ \varrho(r)\, \mathrm{d}r \ ,$$

where

$$a^q(x) \ = \ H(x,r)\frac{\partial}{\partial r}\ln\varrho(r) \ + \ \frac{\partial}{\partial r}H(x,r) \ = \ \left(\begin{array}{c} (x_2+r)/x_1 \\ -1 \end{array} \right)\frac{\partial}{\partial r}\ln\varrho(r) + \left(\begin{array}{c} x_1^{-1} \\ 0 \end{array} \right) \ .$$

Similar to (30), formula (39) in Appendix A implies that an unbiased estimate of the gradient $\nabla_x G(x)$ equals

$$K(\mathbf{q}_\ell(x), S)\, a^{\mathbf{q}_\ell}(x)\,, \tag{31}$$

and

$$\nabla_x G(x) \;=\; \mathbb{E}_\tau[K(\mathbf{q}_\ell(x), S)\, a^{\mathbf{q}_\ell}(x)]\,. \tag{32}$$

A similar estimate can be obtained with the Push Out approach [14]. Thus, we have two unbiased estimates, (29) and (31), for the gradient $\nabla_x G(x)$. For popular distributions such as the exponential or normal distribution, the derivative of the logarithm of the density function, $\frac{\partial}{\partial r}\ln\varrho(r)$, is a constant or a simple function; therefore $a^q(x)$ and the estimate (31) can be easily evaluated. Estimate (29) involves calculation of the density and cumulative distribution functions which may take more time than calculating the estimate (31). Nevertheless, estimate (29) could be preferable because the variance of this estimate is lower than the variance of the estimate (31).

5 Example calculations

This section calculates optimal parameters for an example system. We suppose that the cost and maintenance function per unit of product consists of three terms:

$$c\,(x, S) \;=\; c_1(x_1, x_2) + c_2(S, x_1) + c_3(x_1)\,.$$

The first term is an investment cost for the work station:

$$c_1(x_1, x_2) \;=\; \frac{C_1 x_1}{x_2}\,,$$

the second term is the buffer cost:

$$c_2(S, x_1) \;=\; C_2 S x_1\,,$$

and the third term is the maintenance cost

$$c_3(x_1) \;=\; C_3 x_1\,.$$

Constants C_1, C_2, and C_3 equal:

$$C_1 \;=\; 1\,, \quad C_2 \;=\; 0.1\,, \quad C_3 \;=\; 0.65\,.$$

The gain for the processed batch equals, $\beta = 2$, and the cost of a lost one equals, $\alpha = 3$. The parameter κ in the constraint (14) equals $\kappa = 0.1$ and parameter Q in formula (12) equals $Q = 30$. The mean time to failure of the work station equals $T = 20$. The maximum feasible buffer size S is 8, i.e., $S \in \{S_1, \ldots, S_8\} = \{1, \ldots, 8\}$.

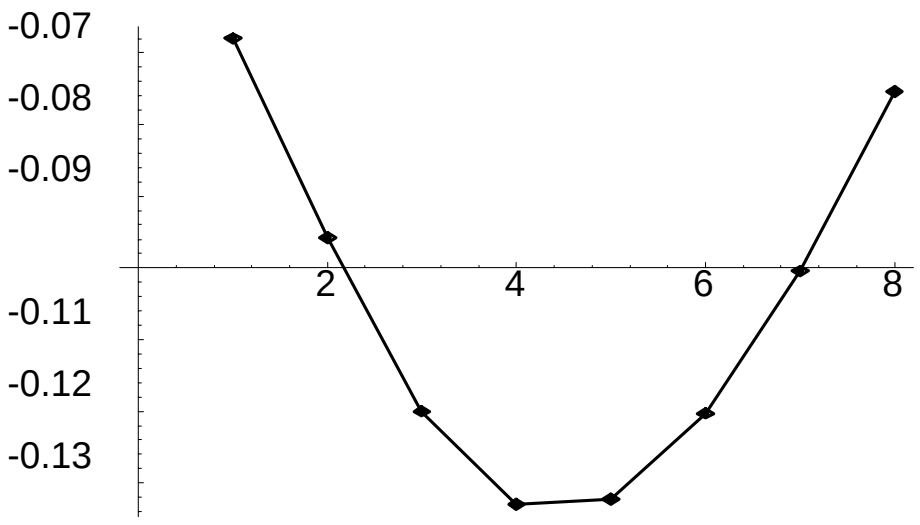

Figure 3: *Optimal values* $C(x_i^*, S_i)$ *for each buffer size* $S_i \in \{1, \ldots, 8\}$.

The repair intervals are normally distributed with parameters $m = 2$, $\sigma = 1$. The normal distribution is truncated at the point 0 to avoid negative values for the repair intervals. Fixing the buffer size S_i reduces problem (13) to a nonlinear minimization problem with linear constraints w.r.t. x . For each buffer size $S_i \in \{1, \ldots, 8\}$, we solved the problem

$$C(x, S_i) \rightarrow \min_{x \in X} \tag{33}$$

w.r.t. x and found an optimum vector x_i^* . We used MATHEMATICA code of the variable metric algorithm [19] running on PC 486. Optimal values for each $S_i \in \{1, \ldots, 8\}$ are plotted in Fig. 3. This figure shows that the optimal buffer size equals $S^* = 4$. The performance function at the optimal point equals $C(x^*, C^*) = -0.133027$ and the optimal vector x^* equals $x^* = (0.47561, 0.37561)$.

References

[1] J.A. Buzacott (1982): "Optimal" operating rules for automated manufacturing systems. *IEEE - Transactions on Automatic Control* **27**, pp. 80-86.

[2] Ermoliev, Yu. (1988): *Stochastic Quasi-Gradient Methods.* In: "Numerical Techniques for Stochastic Optimization" Eds. Yu. Ermoliev and R.J-B Wets, Springer-Verlag,393–401.

[3] Yu. Ermoliev, S. Uryasev, and J. Wessels: (1992): *On Optimization of Dynamical Material Flow Systems Using Simulation.* International Institute for Applied Systems Analysis, Laxenburg, Austria, Report WP-92-76, 28 p.

[4] P. Glasserman (1991): *Gradient Estimation Via Perturbation Analysis.* Kluwer Academic Publishers, Boston-Dordrecht-London.

[5] W.B. Gong, Y.C. Ho (1987) Smoothed (conditional) perturbation analysis of discrete event dynamic systems. *IEEE - Transactions on Automatic Control* **32**, pp. 858-866.

[6] A.M. Gupal, L.T. Bazhenov. (1972): *A Stochastic Analog of the Methods of Conjugate Gradients.* Kibernetika, 1, 124–126, (in Russian).

[7] Ho Y.C. and X.R. Cao (1991): *Perturbation Analysis of Discrete Event Dynamic Systems.* Kluwer, Boston.

[8] M.B.M. de Koster (1989) *Capacity oriented analysis and design of production systems.* Springer-Verlag, Berlin (LNEMS 323).

[9] H.J. Kushner, Hai-Huang. (1981): *Asymptotic Properties of Stochastic Approximation with Constant Coefficients.* SIAM Journal on Control and Optimization, 19, 87–105.

[10] H.J. Kushner, G.G. Jin (1997): *Stochastic Approximation Algorithms and Applications.* Appl. Math. 35, Springer.

[11] Yin G. and Q. Zhang.(Eds) (1996): *Mathematics of Stochastic Manufacturing Systems.* Lectures in Applied Mathematics, Vol. 33.

[12] D. Mitra (1988) *Stochastic fluid models.* in: P.-J. Courtois, G. Latouche (eds.) Performance '87. Elsevier, Amsterdam. pp. 39-51.

[13] Pflug G. Ch. (1996): *Optimization of Stochastic Models, The Interface Between Simulation and Optimization.* Kluwer Academic Publishers, Boston-Dordrecht-London.

[14] Rubinstein, R. and A. Shapiro (1993): *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization via the Score Function Method.* Wiley, Chichester.

[15] Saridis, G.M. (1970): Learning applied to successive approximation algorithms. *IEEE Trans. Syst. Sci. Cybern.*, 1970, SSC-6, Apr., pp. 97–103.

[16] Bashyam, S. and M.S. Fu (1994): Application of Perturbation Analysis to a Class of Periodic Review (s,S) Inventory Systems. *Naval Research Logistics.* Vol. 41, pp.47-80.

[17] Syski, W. (1988): A Method of Stochastic Subgradients with Complete Feedback Stepsize Rule for Convex Stochastic Approximation Problems. *J. of Optim. Theory and Applic.* Vol. **39**, No. **2**, pp. 487–505.

[18] Uryasev, S.P. (1988): *Adaptive Stochastic Quasi–Gradient Methods.* In "Numerical Techniques for Stochastic Optimization". Eds. Yu. Ermoliev and R. J-B Wets. Springer Series in Computational Mathematics **10**, (1988), 373–384.

[19] Uryasev, S. (1991): New Variable-Metric Algorithms for Nondifferential Optimization Problems. *J. of Optim. Theory and Applic.* Vol. 71, No. 2, 1991, 359-388.

[20] Uryasev, S. (1992): A Stochastic Quasi–Gradient Algorithm with Variable Metric, *Annals of Operations Research.* 39, 251-267.

[21] Uryasev, S. (1994): Derivatives of Probability Functions and Integrals over Sets Given by Inequalities. *J. Computational and Applied Mathematics.* Vol. 56, 197-223.

[22] Uryasev, S. (1995): *Derivatives of Probability Functions and Some Applications.* Annals of Operations Research, Vol. 56, 287-311.

[23] Uryasev, S. (1997): Analytic Perturbation Analysis for DEDS with Discontinuous Sample-path Functions. *Stochastic Models.* Vol. 13, No. 3.

[24] J. Wijngaard (1979) The effect of interstage buffer storage on the output of two unreliable production units in series with different production rates. *AIIE - Transactions* **11**, pp. 42-47.

[25] S. Yeralan, E.J. Muth (1987) A general model of a production line with intermediate buffer and station breakdown. *AIIE - Transactions* **19**, pp. 130-139.

6 Appendix A. Description of the analytic perturbation analysis

Let $(\mathbb{P}, \mathcal{F}, \Omega)$ be a probability space, and $G(x) = \mathbb{E}\, g(x, \boldsymbol{\omega})$ be an expectation of the function $g(x, \boldsymbol{\omega})$ depending upon the control variables $x \in \mathbb{R}^n$ and a random element $\boldsymbol{\omega} \in \Omega$.

We suppose that:

1. the set Ω can be split into subsets $\mu^q(x) \in \mathcal{F}$, $q = 1, 2, \ldots$

$$\Omega \;=\; \bigcup_{q=1}^{\infty} \mu^q(x)\;, \tag{34}$$

and the function $g(x, \boldsymbol{\omega})$ is differentiable w.r.t. $\boldsymbol{\omega}$ (or w.r.t. to some components of $\boldsymbol{\omega}$, if $\boldsymbol{\omega}$ is a vector) on each subset $\mu^q(x)$, $q = 1, 2, \ldots$;

2. for any $q \neq j$

$$\mathbb{P}\Big(\mu^q(x) \bigcap \mu^j(x)\Big) \;=\; 0 \;;$$

3. each subset $\mu^q(x)\,,\; q = 1, 2, \ldots$ can be represented by the system of inequalities

$$\mu^q(x) \;=\; \{\, \boldsymbol{\omega} \in \Omega \;:\; f^q(x, \boldsymbol{\omega}) \leq 0 \,\}$$

$$\stackrel{\text{def}}{=} \;\{\, \boldsymbol{\omega} \in \Omega \;:\; f_l^q(x, \boldsymbol{\omega}) \leq 0\,,\; 1 \leq l \leq k^q \,\}\;,$$

where $f_l^q \;:\; \mathbb{R}^n \times \Omega \to \mathbb{R}\,,\; 1 \leq l \leq k^q$.

$I_{\{f^q(x,\boldsymbol{\omega})\leq 0\}}$ denotes an indicator function, which corresponds to the set $\mu^q(x)$

$$I_{\{f^q(x,\boldsymbol{\omega})\leq 0\}} \;=\; \left\{ \begin{array}{l} 1\;, \;\; \text{if}\;, \quad f^q(x, \boldsymbol{\omega}) \leq 0\;; \\ 0\;, \;\; \text{otherwise}\;. \end{array} \right.$$

With these definitions, the performance function $G(x) = \mathbb{E}\, g(x, \boldsymbol{\omega})$ can be represented as the sum

$$G(x) = \mathbb{E}\, g(x, \boldsymbol{\omega}) = \mathbb{E}\Big[\sum_{q=1}^{\infty} I_{\{f^q(x,\boldsymbol{\omega})\leq 0\}}\, g(x, \boldsymbol{\omega})\Big] = \sum_{q=1}^{\infty} \mathbb{E}\Big[I_{\{f^q(x,\boldsymbol{\omega})\leq 0\}}\, g(x, \boldsymbol{\omega})\Big]\;. \quad (35)$$

The function

$$\phi_q(x) \;\stackrel{\text{def}}{=}\; \mathbb{E}\Big[I_{\{f^q(x,\boldsymbol{\omega})\leq 0\}}\, g(x, \boldsymbol{\omega})\Big]$$

is an integral of the function $g(x, \boldsymbol{\omega})$ over the set $\mu^q(x)$. This function is differentiable under general conditions (see formulas (41), (43), and (44)). The gradient $\nabla_x \phi_q(x)$ can be represented as an integral of another function $a^q(x, \boldsymbol{\omega})$ over the same set $\mu^q(x)$ plus an additional function $\psi_q(x)$, which is a surface integral, or equivalently, it can be represented as a mathematical expectation of the product $a^q(x, \boldsymbol{\omega}) I_{\{f^q(x,\boldsymbol{\omega})\leq 0\}}$, plus $\psi_q(x)$, i.e.,

$$\nabla_x \phi_q(x) \;=\; \mathbb{E}\Big[I_{\{f^q(x,\boldsymbol{\omega})\leq 0\}}\, a^q(x, \boldsymbol{\omega})\Big] \;+\; \psi^q(x)\;. \quad (36)$$

The function $\phi_q(x)$ can be differentiated directly, or it can be written as

$$\phi_q(x) \;=\; \mathbb{E}\Big[I_{\{f^q(x,\boldsymbol{\omega})\leq 0\}}\, g(x, \boldsymbol{\omega})\Big] \;=\; \mathbb{E}\Big[\mathbb{E}_q\Big[I_{\{f^q(x,\boldsymbol{\omega})\leq 0\}}\, g(x, \boldsymbol{\omega})\Big]\Big]\;,$$

where $\mathbb{E}_q$ is a conditional expectation such that the function

$$\mathbb{E}_q\Big[I_{\{f^q(x,\boldsymbol{\omega})\leq 0\}}\, g(x, \boldsymbol{\omega})\Big] \quad (37)$$

is smooth w.r.t. x. Further, we can interchange the gradient and expectation signs

$$\nabla_x \phi_q(x) \;=\; \nabla_x \mathbb{E}\Big[\mathbb{E}_q\Big[I_{\{f^q(x,\boldsymbol{\omega})\le 0\}}\, g(x,\boldsymbol{\omega})\Big]\Big] \;=\; \mathbb{E}\Big[\nabla_x \mathbb{E}_q\Big[I_{\{f^q(x,\boldsymbol{\omega})\le 0\}}\, g(x,\boldsymbol{\omega})\Big]\Big]\;,$$

and apply differentiation formulas (41), (43), and (44) to the function (37). Thus, (35) and (36) imply

$$\begin{aligned}
\nabla_x G(x) \;&=\; \nabla_x \sum_{q=1}^{\infty} \phi_q(x) \;=\; \sum_{q=1}^{\infty} \nabla_x \phi_q(x) \\
&=\; \sum_{q=1}^{\infty} \mathbb{E}\Big[I_{\{f^q(x,\boldsymbol{\omega})\le 0\}}\, a^q(x,\boldsymbol{\omega})\Big] \;+\; \sum_{q=1}^{\infty} \psi^q(x) \\
&=\; \mathbb{E}\, a^{\mathbf{q}}(x,\boldsymbol{\omega}) \;+\; \sum_{q=1}^{\infty} \psi^q(x)\;, \qquad (38)
\end{aligned}$$

where,

$$\mathbf{q} \;=\; \min\{\nu : \boldsymbol{\omega} \in \mu^{\nu}(x)\}\;.$$

If $\sum_{q=1}^{\infty} \psi^q(x) \;=\; 0$, then $a^{\mathbf{q}}(x,\boldsymbol{\omega})$ is an unbiased estimate of the gradient, which can be evaluated together with the sample-path function $g(x,\boldsymbol{\omega})$ during the same simulation run. If $\sum_{q=1}^{\infty} \psi^q(x) \neq 0$, it is desirable to find an expression for this term so that it can be calculated during the same simulation run, together with $g(x,\boldsymbol{\omega})$ and $a^{\mathbf{q}}(x,\boldsymbol{\omega})$. In some cases, it is possible to convert $\psi^q(x)$ using artificial variables to the integral over the volume

$$\psi^q(x) \;=\; \mathbb{E}\Big[I_{\{f^q(x,\boldsymbol{\omega})\le 0\}}\, b^q(x,\boldsymbol{\omega})\Big]\;.$$

Then, with (38),

$$\nabla_x G(x) \;=\mathbb{E}\, a^{\mathbf{q}}(x,\boldsymbol{\omega}) \;+\; \sum_{q=1}^{\infty} \mathbb{E}\Big[I_{\{f^q(x,\boldsymbol{\omega})\le 0\}}\, b^q(x,\boldsymbol{\omega})\Big] \;=\; \mathbb{E}[a^{\mathbf{q}}(x,\boldsymbol{\omega}) + b^{\mathbf{q}}(x,\boldsymbol{\omega})]\;. \qquad (39)$$

Thus, $d^{\mathbf{q}}(x,\boldsymbol{\omega}) \;\triangleq\; a^{\mathbf{q}}(x,\boldsymbol{\omega}) + b^{\mathbf{q}}(x,\boldsymbol{\omega})$ is an unbiased estimate of the gradient. The random vector $d^{\mathbf{q}}(x,\boldsymbol{\omega})$ can be obtained with one Monte-Carlo simulation run of the model, analogous to the random value $g(x,\boldsymbol{\omega})$.

The estimate $d^{\mathbf{q}}(x,\boldsymbol{\omega})$ can be used in Monte-Carlo type simulations. Standard variance reduction techniques (see, for example [13]), such as conditioning, coupling, and importance sampling can be used to reduce the variance of this estimate. Let measure $\rho(x,\omega)$ dominate the measure $p(x,\omega)$. Importance sampling technique, changing the measure in expectation

$$\mathbb{E}[d^{\mathbf{q}}(x,\boldsymbol{\omega})] \;\triangleq\; \int d^{q(\omega)}(x,\omega)\, p(x,\omega)\, \mathrm{d}\omega \;=$$

$$\int d^{q(\omega)}(x,\omega)\,\frac{p(x,\omega)}{\rho(x,\omega)}\rho(x,\omega)\,\mathrm{d}\omega \;=\; \mathbb{E}_\rho[d^{\mathbf{q}}(x,\boldsymbol{\omega})\frac{p(x,\boldsymbol{\omega})}{\rho(x,\boldsymbol{\omega})}]\;,$$

may significantly improve the variance.

7 Appendix B. Analytical derivatives of the integrals over sets given by inequalities

Let the function

$$F(x) = \int\limits_{f(x,y)\,\leq\,0} p(x,y)\,\mathrm{d}y \tag{40}$$

be defined on the Euclidean space $\mathbb{R}^n$, where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^k$ and $p : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ are some functions. The inequality $f(x,y) \leq 0$ should be treated as a system of inequalities

$$f_i(x,y) \leq 0\;,\quad i = 1,\ldots,k\;.$$

Further, we presented a general formula [21, 22] for the differentiation of integral (40). A gradient of the integral is represented as a sum of integrals taken over a volume and over a surface.

Let us introduce the following shorthand notations

$$f(x,y) = \begin{pmatrix} f_1(x,y) \\ \vdots \\ f_k(x,y) \end{pmatrix}\;,\quad f_{1l}(x,y) = \begin{pmatrix} f_1(x,y) \\ \vdots \\ f_l(x,y) \end{pmatrix}\;,$$

$$\nabla_y f(x,y) = \begin{pmatrix} \frac{\partial f_1(x,y)}{\partial y_1}\;, & \ldots & , \frac{\partial f_k(x,y)}{\partial y_1} \\ & \vdots & \\ \frac{\partial f_1(x,y)}{\partial y_m}\;, & \ldots & , \frac{\partial f_k(x,y)}{\partial y_m} \end{pmatrix}\;.$$

Divergence for the matrix H is defined as

$$\mathrm{div}_y\, H = \begin{pmatrix} \sum\limits_{i=1}^{m} \frac{\partial h_{1i}}{\partial y_i} \\ \vdots \\ \sum\limits_{i=1}^{m} \frac{\partial h_{ni}}{\partial y_i} \end{pmatrix}\;,\quad H = \begin{pmatrix} h_{11}\;, & \ldots & , h_{1m} \\ & \vdots & \\ h_{n1}\;, & \ldots & , h_{nm} \end{pmatrix}\;.$$

We define

$$\mu(x) = \{\, y \in \mathbb{R}^m \,:\, f(x,y) \le 0 \,\} \stackrel{\text{def}}{=} \{\, y \in \mathbb{R}^m \,:\, f_l(x,y) \le 0\,,\ 1 \le l \le k \,\}\,,$$

and $\partial\mu(x)$ to be the surface of the set $\mu(x)$. Let us denote by $\partial_i\mu(x)$ a part of the surface which corresponds to the function $f_i(x,y)$

$$\partial_i\mu(x) \;=\; \mu(x) \bigcap \{\, y \in \mathbb{R}^m \,:\, f_i(x,y) = 0 \,\}\,.$$

If we split the set $K \stackrel{\text{def}}{=} \{1,\ldots,k\}$ into two subsets K_1 and K_2, without a loss of generality, we can consider

$$K_1 \;=\; \{1,\ldots,l\} \;\text{ and }\; K_2 \;=\; \{l+1,\ldots,k\}\,.$$

There is freedom in the choice of the sets K_1 and K_2 and representation of the gradient of function (40). First, we consider the case when the subsets K_1 and K_2 are not empty. In this case, the derivative of integral (40) is given by the formula

$$\nabla_x F(x) \;=\; \int\limits_{\mu(x)} \Big[\, \nabla_x p(x,y) \;+\; \mathrm{div}_y\Big(p(x,y) H_l(x,y)\Big) \,\Big]\, \mathrm{d}y \;-$$

$$-\; \sum_{i=l+1}^{k} \int\limits_{\partial_i\mu(x)} \frac{p(x,y)}{||\nabla_y f_i(x,y)||} \Big[\, \nabla_x f_i(x,y) \;+\; H_l(x,y)\, \nabla_y f_i(x,y) \,\Big]\, \mathrm{d}S\,, \qquad (41)$$

where the matrix function $H_l : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^{n\times m}$ satisfies the equation

$$H_l(x,y)\, \nabla_y f_{1l}(x,y) \;+\; \nabla_x f_{1l}(x,y) \;=\; 0\,. \qquad (42)$$

The last equation can have a lot of solutions and we can choose an arbitrary solution, differentiable with respect to the variable y .

Further, let us present the derivative of function (40) for the case with the empty set K_1. Then, matrix function H_l is absent and

$$\nabla_x F(x) \;=\; \int\limits_{\mu(x)} \nabla_x p(x,y)\, \mathrm{d}y \;-\; \sum_{i=1}^{k} \int\limits_{\partial_i\mu(x)} \frac{p(x,y)}{||\nabla_y f_i(x,y)||} \nabla_x f_i(x,y)\, \mathrm{d}S\,. \qquad (43)$$

Finally, let us consider a formula for the derivative of function (40) for the case with the empty set K_2. The integral over the surface is absent and the derivative is represented as an integral over the volume

$$\nabla_x F(x) \;=\; \int\limits_{\mu(x)} \Big[\, \nabla_x p(x,y) \;+\; \mathrm{div}_y\Big(p(x,y) H(x,y)\Big) \,\Big]\, \mathrm{d}y\,, \qquad (44)$$

where a matrix function $H : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^{n \times m}$ satisfies the equation

$$H(x,y)\,\nabla_y f(x,y) \;+\; \nabla_x f(x,y) \;=\; 0\;. \tag{45}$$

In many cases, there is a simple way to solve equations (42) and (45) using a change of variables. Suppose that there is a change of variables

$$y = \gamma(x,z)$$

which eliminates vector x from the function $f(x,y)$, i.e., function $f(x,\gamma(x,z))$ does not depend upon variable x. Denote by $\gamma^{-1}(x,y)$ the inverse function, defined by the equation

$$\gamma^{-1}\Big(x,\gamma(x,z)\Big) \;=\; z\;.$$

In this case, the matrix

$$H(x,y) \;=\; \nabla_x \gamma(x,z)|_{z=\gamma^{-1}(x,y)}\;. \tag{46}$$

is a solution of equation (45).