Optimal Constrained Design of Control Charts Using Stochastic Approximations

Abstract

In statistical process monitoring, control charts typically depend on a set of tuning parameters besides its control limit(s). Proper selection of these tuning parameters is crucial to their performance. In a specific application, a control chart is often designed for detecting a target process distributional shift. In such cases, the tuning parameters should be chosen such that some characteristic of the out-of-control (OC) run length of the chart, such as its average, is minimized for detecting the target shift, while the control limit is set to maintain a desired in-control (IC) performance. However, explicit solutions for such a design are unavailable for most control charts, and thus numerical optimization methods are needed. In such cases, Monte Carlo-based methods are often a viable alternative for finding suitable design constants. The computational cost associated with such scenarios is often substantial, and thus computational efficiency is a key requirement. To address this problem, a two-step design based on stochastic approximations is presented in this paper, which is shown to be much more computationally efficient than some representative existing methods. A detailed discussion about the new algorithm's implementation along with some examples are provided to demonstrate the broad applicability of the proposed methodology for the optimal design of univariate and multivariate control charts. Computer codes in the Julia programming language are also provided in the supplemental material.

Keywords: Statistical process monitoring; Tuning parameters; Numerical optimization; Stochastic approximations; Simultaneous perturbations.

1 Introduction

Statistical Process Control (SPC) involves monitoring one or more quality variables of a process over time, and control charts are typically employed to signal shifts in one or more process parameters (Qiu, 2013). The performance of a control chart is usually measured in terms of the run length (RL), which is the number of observation times required to detect a shift. The charting statistic of a control chart is compared to control limit(s) chosen to ensure that some characteristics of the RL, such as the mean or its quantiles, have certain pre-specified properties when the process is in-control (IC). Then, the control chart is designed by choosing its tuning parameters so that a pre-specified target shift can be detected in the quickest way. Two common performance metrics employed in SPC are the IC average run length (ARL_{IC}) and the out-of-control (OC) average run length (ARL_{OC}). Typically, ARL_{IC} is pre-specified at a nominal level, denoted as ARL_0 hereafter, and the control chart is then designed to minimize ARL_{OC} for detecting a pre-specified target shift in the quality variables.

Minimization of metrics based on the OC run length is not only relevant in applications, but also important in simulation studies for comparing the performance of different control charts. Thus, efficient optimization algorithms are critically important to ensure feasibility of their computation. **In cases to use the CUSUM control chart for monitoring the mean of Gaussian data (Page, 1954), the optimal design of the chart is known (i.e., its allowance constant can be set to be half of a target shift). In cases with some other basic control charts for monitoring processes with specific parametric distributions, the optimal design problem can be solved using numerical approaches. For instance, when the conventional EWMA control chart is used for monitoring normally-distributed processes, the Markov Chain approximation methods (Brook and Evans, 1972; Lucas and Saccucci, 1990; Jones, 2002; Capizzi and Masarotto, 2003; Wang et al., 2021) are often employed. Alternatively, the integral equation approach (Page, 1954) can also be**

used, in which a recurrence relation for the average run length of the control chart can be derived and numerical methods such as the Gaussian quadrature procedure can be utilized to solve the integral equation (Vance, 1986; Crowder, 1987; Fellner, 1990; Rigdon, 1995b,a; Capizzi and Masarotto, 2010).

However, in many other cases, the optimal design problem can be analytically and numerically intractable due to the complexity of either the control chart or the process under monitoring. As a result, people often use the alternative method to approximate the performance metrics by Monte Carlo simulations. The simulation-based approach transforms the optimal design of a control chart into a stochastic optimization problem. There are some different methods of stochastic optimization in the literature, and one promising approach is based on Stochastic Approximations (SA). SA methods are a powerful class of algorithms to optimize noisy functions, wherein the exact value of the function cannot be calculated directly, but is observed indirectly in the presence of stochastic noise. Initially proposed by Robbins and Monro (1951), SA methods have been subject to numerous developments, and there exists an extensive literature that provides examples of their applications in adaptive control and nonlinear optimization (Ruppert, 1991; Lai, 2003; Kushner and Yin, 2003; Spall, 2003).

SA methods in the context of SPC have been studied by Yashchin (1993), who discussed the computational difficulties in applying the Robbins-Monro algorithm (Robbins and Monro, 1951). Recently, SA methods in SPC have been shown highly efficient in determining the control limits of a control chart so that either ARL_{IC} or some alternative quantiles of the IC run length achieve a desired level (Capizzi and Masarotto, 2009, 2016). So far, the SA-based approaches have only been used to determine the control limits of a chart when its tuning parameters are pre-specified. To the extent of our knowledge, there is no SA-based algorithm to optimize the tuning parameters so that a target process shift can be detected by the chart in the fastest way. This paper aims to fill the gap.

Current methods in the literature to find the optimal tuning parameters are through Monte Carlo simulations by first finding appropriate control limit values for each set of

values of the tuning parameters and then searching for the optimal values of the tuning parameter to optimize a specific characteristic of the OC run length using traditional searching approaches for function minimization. Some examples include grid search (Qiu, 2008) and numerical solvers (Mahmoud and Zahran, 2010). However, these methods are not designed specifically to handle noisy functions, and thus may not be able to provide an accurate estimation of the optimal solution. An algorithm that leverages the noisiness of the RL function could be more efficient in determining the optimal tuning parameters. Motivated by this intuition, we propose a two-step SA-based optimization algorithm that combines the SA method of Capizzi and Masarotto (2016) and the Simultaneous Perturbation Stochastic Approximation (SPSA) method of Spall (1992). The proposed method can estimate the optimal tuning parameters for detecting a given target shift, while satisfying a constraint on some characteristics of the IC run length, for both univariate and multivariate control charts. It provides a flexible optimization algorithm that is computationally efficient.

SA-based methods aim to estimate the gradient of an objective function by Monte Carlo simulations. Estimation of the gradient in the context of control chart design has been explored previously by other researchers. For instance, Shu et al. (2014) and Huang et al. (2016) used gradient information to solve the integral equations of the EWMA and CUSUM control charts, respectively. Huang et al. (2018) considered the optimal design of a MEWMA control chart using the Markov chain approximations. Our proposed method differs from these previous methods in that we use Monte Carlo simulations, rather than numerical approaches, to estimate the gradient. This allows our method to be applicable to a wide range of control charts for monitoring processes with various distributions.

The remainder of the article is organized as follows. In Section 2, the framework of SPC is introduced briefly, and the main theory behind the SPSA algorithm is also discussed. The proposed two-step SA algorithm is then introduced in detail, including some practical

guidelines for choosing its parameters and a convergence criterion established based on its theoretical properties. In Section 3, the SPSA algorithm is illustrated using the classical EWMA control chart (Roberts, 1959) and the recently proposed R-SADA control chart (Xian et al., 2019) for detecting mean shifts when process quality variables are partially observed. The latter control chart is considered to show that the proposed method can be used in both standard and non-standard settings. Although mean shifts are considered in all examples in this paper, the proposed method is actually general and can be used in cases for detecting shifts in other process characteristics such as variance and skewness. In Section 4, the accuracy and computational efficiency of the proposed algorithm is compared to two traditional approaches, including the grid search (Qiu, 2008) and Nelder-Mead (Nelder and Mead, 1965) algorithms. In the comparison, all methods are applied to the problem of choosing the optimal allowance constant of the classical CUSUM chart (Page, 1954). Since an explicit solution is available in this problem, it can be used as a benchmark to evaluate the performance of the related algorithms. Finally, Section 5 provides some concluding remarks. Computer code in Julia implementing the proposed algorithm is available as part of the supplemental material.

2 SPSA Optimization of the Tuning Parameters

2.1 The problem

Let $\{\boldsymbol{X}_t, t \geq 1\}$ be observations of a p-dimensional process, and $C_t = C_t(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t; \boldsymbol{\zeta}) \in \mathbb{R}$ be the charting statistic of a control chart computed from the historic data $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}$ at the current time t, where $\boldsymbol{\zeta} \in \mathcal{Z} \subseteq \mathbb{R}^d$ is a d-dimensional vector of tuning parameters. The charting statistic is then compared to a control limit $h = h(\boldsymbol{\zeta})$ which may depend on the tuning parameters $\boldsymbol{\zeta}$, and an OC alarm is triggered if $|C_t| > h(\boldsymbol{\zeta})$. Without loss of generality, the two-sided version of the control chart is discussed here, and the one-sided version can be discussed in the same way. Then, the run length (RL) of the chart is defined

to be

$$\mathrm{RL}[\boldsymbol{\zeta}, h(\boldsymbol{\zeta})] = \inf\{t > 0 : |C_t| > h(\boldsymbol{\zeta})\}. \tag{1}$$

Namely, $\mathrm{RL}[\boldsymbol{\zeta}, h(\boldsymbol{\zeta})]$ is the first time when an OC alarm is triggered. For a given $\boldsymbol{\zeta}$, practitioners typically find the value of the control limit $h(\boldsymbol{\zeta})$ such that

$$\mathrm{ARL}_{\mathrm{IC}} = \mathbb{E}_0\{\mathrm{RL}[\boldsymbol{\zeta}, h(\boldsymbol{\zeta})]\} = \mathrm{ARL}_0, \tag{2}$$

where $\mathbb{E}_0\{\cdot\}$ denotes the expectation under the assumption that the process under monitoring is IC, and ARL_0 is a desired value of the IC average run length.

To be more specific, let us consider two classical control charts. First, the CUSUM chart (Page, 1954) for detecting upward mean shifts in a Normal process $\{X_t \sim N(0,1), t \geq 1\}$ has its charting statistic at time t defined to be

$$C_t = \max\{0, C_{t-1} + X_t - k\}, \quad \text{for } t \geq 1,$$

where $C_0 = 0$, and $\zeta = k$ is the tuning parameter which is also known as the allowance constant. Then, the chart raises an alarm whenever $C_t > h(k)$, where $h(k) > 0$ is a control limit. This chart assumes that the IC process observations at different times are independent and identically distributed (i.i.d.). Second, the Multivariate Exponentially Weighted Moving Average (MEWMA) chart (Lowry et al., 1992) is designed to monitor the mean of a p-dimensional process $\{\boldsymbol{X}_t, t \geq 1\}$, where the IC process observations are assumed to be i.i.d. with mean $\boldsymbol{\mu}$ and covariance matrix Σ. Its charting statistic is defined to be

$$T_t^2 = \boldsymbol{Z}_t' \Sigma_{\boldsymbol{Z}_t}^{-1} \boldsymbol{Z}_t, \quad \text{for } t \geq 1, \tag{3}$$

where $\boldsymbol{Z}_t = (I - \Lambda)\boldsymbol{Z}_{t-1} + \Lambda(\boldsymbol{X}_t - \boldsymbol{\mu})$ for $t \geq 1$, and $\boldsymbol{Z}_0 = \boldsymbol{0}$, $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_p)$ is the weighting matrix. The MEWMA chart gives a signal of mean shift if $T_t^2 > h(\lambda_1, \ldots, \lambda_d)$, where $h(\lambda_1, \ldots, \lambda_d) > 0$ is the control limit. In this example, the vector of tuning parameters is $\boldsymbol{\zeta} = (\lambda_1, \lambda_2, \ldots, \lambda_p) \in (0, 1]^p$.

The problem of finding the optimal value of the tuning parameters, denoted as $\boldsymbol{\zeta}^*$, for detecting an OC scenario is most commonly formulated in terms of minimizing ARL_{OC}, under a constraint on ARL_{IC}. Namely, we want to find $\boldsymbol{\zeta}^* \in \mathcal{Z}$ such that

$$\begin{aligned} \boldsymbol{g}(\boldsymbol{\zeta}^*) = \frac{\partial \mathbb{E}_1\{\text{RL}[\boldsymbol{\zeta}, h(\boldsymbol{\zeta})]\}}{\partial \boldsymbol{\zeta}}\Big|_{\boldsymbol{\zeta}=\boldsymbol{\zeta}^*} &= \mathbf{0}, \\ \text{s.t.} \quad \mathbb{E}_0\{\text{RL}[\boldsymbol{\zeta}^*, h(\boldsymbol{\zeta}^*)]\} &= \text{ARL}_0, \end{aligned} \tag{4}$$

where $\mathbb{E}_1\{\cdot\}$ denotes the OC expectation after the process becomes OC. Note that, due to the stochastic nature of the problem, the expected values in (4) usually do not have a closed form, and can only be approximated via simulation.

For the optimization problem (4), its optimum $\boldsymbol{\zeta}^*$ exists in $\mathcal{Z}$ in cases when conventional control charts (e.g., CUSUM and EWMA charts) are used for monitoring processes with certain commonly-used parametric distributions in the exponential distribution family since it can be checked that the related objective functions would be convex. However, in cases with complex control charts and data distributions, such analytical results would not be possible. However, these complex situations are also the ones in which Monte-Carlo methods are especially useful. In such cases, although it may not be possible to guarantee the existence of a global optimum, Monte-Carlo methods can always find a local minimum instead to achieve a "reasonably good" value of the objective function, which is often appropriate for practical purposes.

2.2 Some generalizations

It is worth noting that the SPC framework described by Equations (1) and (2) does not include all applications of process monitoring. For instance, it excludes scenarios involving multiple control charts being used simultaneously. This typically requires the specification of different target shifts for choosing process parameters of individual control charts that must then be considered jointly in the optimization. This situation represents a case of

multi-objective optimization (Branke et al., 2008), wherein multiple objective functions need to be minimized simultaneously. Various strategies can be employed to obtain a satisfactory solution in such cases. One viable approach involves the use of a weighting mechanism to aggregate the objective functions, enabling a combined optimization that takes into account the relative importance of each objective function. An alternative method is to employ a lexicographic optimization approach, whereby the functions are optimized in a sequential manner, with each solution serving as a soft constraint for the subsequent objective. The second approach is similar to the adaptive EWMA (AEWMA) control chart which optimizes its weighting parameter by considering both a small and a large location shift (see Capizzi and Masarotto, 2003). However, finding an appropriate general approach for the multi-chart tuning parameter optimization is out the scope of the current paper, and thus left as a topic for future research.

Additionally, note that the optimization problem (4) is defined using ARL_{OC} and the ARL_{IC}, since they are the most widely utilized metrics for evaluating the performance of control charts. However, alternative choices are possible for both the objective function and the constraint. For instance, one could also consider quantiles of the OC and IC RL distributions, such as the median OC RL (MRL_{OC}) and the median IC RL (MRL_{IC}) in place of ARL_{OC} and ARL_{IC}, respectively. To accommodate this alternative setting, a modification of the proposed SPSA algorithm can be introduced, as discussed in Section 2.4.1. Furthermore, incorporating a constraint on the quantiles of the IC RL distribution can be achieved after a straightforward adjustment, as discussed in Section 2.4.4.

The optimal design using (4) can be further generalized to accommodate some additional information. For example, under the economic design framework Duncan (1956); Lorenzen and Vance (1986); Ho and Case (1994), the sample size and sampling interval of the process observations can also be adjusted, besides the selection of the control limit. In such cases, a loss function can be defined as a function of all these quantities. Some researchers have also suggested using an indifference region to account for small shifts that may not

have practical relevance (Woodall, 1985; Aparisi and García-Díaz, 2007; Kuiper and Goedhart, 2023). This indifference region allows for a certain degree of tolerance when monitoring a process. In addition, a distribution on the magnitude of the anticipated process shift can be introduced to describe uncertainty in the anticipated shift (Chen and Chen, 2007; Ryu et al., 2010). In such cases, the expected value of ARL_{OC} with respect to the prior distribution of the shift can be minimized. Although it is possible to generalize our proposed method to these cases, the generalizations may be nontrivial and are left for future research.

2.3 The proposed SPSA optimization procedure

Let $Q(\boldsymbol{\zeta}, h(\boldsymbol{\zeta}))$ and $K(\boldsymbol{\zeta}, h(\boldsymbol{\zeta}))$ be two score functions such that the minimization problem can be written as

$$\begin{aligned} \boldsymbol{g}(\boldsymbol{\zeta}^*) &= \frac{\partial Q[\boldsymbol{\zeta}, h(\boldsymbol{\zeta}))]}{\partial \boldsymbol{\zeta}}\Big|_{\boldsymbol{\zeta}=\boldsymbol{\zeta}^*} = \mathbf{0}, \\ \text{s.t. } & K[\boldsymbol{\zeta}^*, h(\boldsymbol{\zeta}^*)] = b, \end{aligned} \tag{5}$$

where b is a pre-specified value for the constraint. To exemplify, equation (4) is a special case of (5) when choosing $Q(\boldsymbol{\zeta}, h(\boldsymbol{\zeta})) = \mathbb{E}_1\{\text{RL}[\boldsymbol{\zeta}, h(\boldsymbol{\zeta})]\}$, $K(\boldsymbol{\zeta}, h(\boldsymbol{\zeta})) = \mathbb{E}_0\{\text{RL}[\boldsymbol{\zeta}, h(\boldsymbol{\zeta})]\}$, and $b = \text{ARL}_0$. The variability in Q and K refers to the Monte Carlo variability when generating RL values by either a) sampling data from the true OC and IC RL distributions when they are assumed known, or b) approximating the OC and IC RL distributions using methods such as the bootstrap (e.g., Gandy and Kvaløy, 2013).

SA methods are commonly used in stochastic optimization (cf., Spall, 2003), and designed to solve the problem (5) using the following iterative procedure:

$$\widehat{\boldsymbol{\zeta}}_{k+1} = \Psi\left(\widehat{\boldsymbol{\zeta}}_k - a_k \frac{\partial Q[\boldsymbol{\zeta}, h(\boldsymbol{\zeta})]}{\partial \boldsymbol{\zeta}}\Big|_{\boldsymbol{\zeta}=\widehat{\boldsymbol{\zeta}}_k}\right), \quad k \geq 1, \tag{6}$$

where Ψ is a projection that maps $\mathbb{R}^d$ to the nearest point in $\mathcal{Z}$, and $a_k > 0$, for $k \geq 1$, is called a gain sequence of the gradient, whose selection will be discussed in Section 2.4.2.

The choice of Ψ depends on the particular control chart considered. For instance, when optimizing the weighting parameter $\lambda \in \mathcal{Z} = [0, 1]$ of an EWMA control chart, the transformation $\Psi(\lambda) = \min(1, \max(0, \lambda))$ can constrain the tuning parameter in the set $\mathcal{Z}$. The major limitation of (6) is that the gradient is typically unavailable in a closed form. A solution to overcome this limitation is to substitute the true gradient with its estimate in the current iteration by considering

$$\widehat{\boldsymbol{\zeta}}_{k+1} = \Psi\left(\widehat{\boldsymbol{\zeta}}_k - a_k \widehat{\boldsymbol{g}}_k(\widehat{\boldsymbol{\zeta}}_k)\right), \quad k \geq 1, \tag{7}$$

where $\widehat{\boldsymbol{g}}_k$ is obtained by sampling values of the score function Q in a neighborhood of the current estimate $\widehat{\boldsymbol{\zeta}}_k$.

To estimate the gradient in (7), one efficient solution is provided by the SPSA algorithm (Spall, 1992, 1998). The key idea is to use a random perturbation $\boldsymbol{\Delta}_k = (\Delta_{1k}, \ldots, \Delta_{dk})^\top$, where Δ_{jk}'s are independent zero-mean random variables that are symmetric about zero and uniformly bounded for all $j = 1, \ldots, d$ and $k = 1, 2, \ldots$ (Spall, 2003). The perturbation $\boldsymbol{\Delta}_k$ is then used in the current iteration to obtain the perturbed estimates $\widehat{\boldsymbol{\zeta}}_k^+ = \Psi(\widehat{\boldsymbol{\zeta}}_k + c_k\boldsymbol{\Delta}_k)$ and $\widehat{\boldsymbol{\zeta}}_k^- = \Psi(\widehat{\boldsymbol{\zeta}}_k - c_k\boldsymbol{\Delta}_k)$, where c_k, for $k = 1, 2, \ldots$, is the gain sequence of the perturbation that will be discussed in Section 2.4.2. Then, it can be shown (cf., Spall, 1992) that the Simultaneous Perturbation (SP) gradient estimate

$$\widehat{\boldsymbol{g}}_k(\widehat{\boldsymbol{\zeta}}_k) = \frac{Q[\widehat{\boldsymbol{\zeta}}_k^+, h(\widehat{\boldsymbol{\zeta}}_k^+)] - Q[\widehat{\boldsymbol{\zeta}}_k^-, h(\widehat{\boldsymbol{\zeta}}_k^-)]}{2c_k} \begin{pmatrix} \Delta_{1k}^{-1} \\ \vdots \\ \Delta_{dk}^{-1} \end{pmatrix}, \tag{8}$$

is a first-order unbiased estimate of the true gradient $\boldsymbol{g}$ at $\widehat{\boldsymbol{\zeta}}_k$ (Spall, 2003, pages 179-180). It is common to choose the perturbations to be the following independent symmetric

Rademacher random variables (Hitczenko and Kwapień, 1994):

$$\Delta_{jk} = \begin{cases} 1 & \text{with probability } 1/2, \\ -1 & \text{with probability } 1/2. \end{cases}, \quad \Delta_{jk} \perp\!\!\!\perp \Delta_{ik} \text{ if } i \neq j, \text{ for } i, j = 1, \ldots, d. \tag{9}$$

It is obvious that $\Delta_{jk}^{-1} = \Delta_{jk}$, for any j and k. To compute the gradient (8), we should compute the two control limits $h(\widehat{\boldsymbol{\zeta}}^{+})$ and $h(\widehat{\boldsymbol{\zeta}}^{-})$ that satisfy the constraint in (5) in advance in each perturbed iteration. This is the most computationally demanding step in the gradient evaluation process. An efficient strategy is described in Section 2.4.4.

Once $h(\widehat{\boldsymbol{\zeta}}^{+})$ and $h(\widehat{\boldsymbol{\zeta}}^{-})$ have been determined, it requires little computational effort to compute multiple values of the score function Q, since Q is usually based on some simulated OC run lengths that are generally much shorter than those under the IC condition when the control chart is unbiased (Pignatiello Jr et al., 1995; Knoth and Morais, 2015). To take advantage of this, we suggest a modification of the SP gradient estimation procedure to enhance its stability and the convergence of the related iterative algorithm. This modification involves substituting $\widehat{\boldsymbol{g}}_k(\widehat{\boldsymbol{\zeta}}_k)$ with

$$\bar{\boldsymbol{g}}_k(\widehat{\boldsymbol{\zeta}}_k) = \frac{\overline{Q}\left[\widehat{\boldsymbol{\zeta}}_k^{+}, h(\widehat{\boldsymbol{\zeta}}_k^{+})\right] - \overline{Q}\left[\widehat{\boldsymbol{\zeta}}_k^{-}, h(\widehat{\boldsymbol{\zeta}}_k^{-})\right]}{2c_k} \begin{pmatrix} \Delta_{1k}^{-1} \\ \vdots \\ \Delta_{dk}^{-1} \end{pmatrix}, \tag{10}$$

where $\overline{Q}[\boldsymbol{\zeta}, h(\boldsymbol{\zeta})]$ is a quantity based on r independent simulations of the score function Q. See Section 2.4.1 for the definition of $\overline{Q}[\boldsymbol{\zeta}, h(\boldsymbol{\zeta})]$ and a discussion about the recommended value of r. This modification is mentioned in Spall (1998) as a way to improve the accuracy of the gradient estimate, reduce the variability, and accelerate the convergence of the algorithm at a relatively small additional computational cost. In addition, to minimize the variability in the differences $\overline{Q}[\widehat{\boldsymbol{\zeta}}_k^{+}, h(\widehat{\boldsymbol{\zeta}}_k^{+})] - \overline{Q}[\widehat{\boldsymbol{\zeta}}_k^{-}, h(\widehat{\boldsymbol{\zeta}}_k^{-})]$ due to random noise, it is recommended to set the seeds of the random number generator in each pair of the simulated score evaluations $\{Q_l[\widehat{\boldsymbol{\zeta}}_k^{+}, h(\widehat{\boldsymbol{\zeta}}_k^{+})], Q_l[\widehat{\boldsymbol{\zeta}}_k^{-}, h(\widehat{\boldsymbol{\zeta}}_k^{-})]\}$ to be the same, for all $l = 1, \ldots, r$. This

is a standard approach used in stochastic optimization to improve the robustness of the optimization (see Kushner and Yin, 2003). An additional enhancement to reduce the variability of the final estimate is to consider the Polyak-Ruppert averaging (Ruppert, 1991; Polyak and Juditsky, 1992). Namely, the following running average is used as the estimate of $\boldsymbol{\zeta}^*$:

$$\bar{\boldsymbol{\zeta}}_k = \frac{1}{k - N_{\mathrm{f}}} \sum_{\ell=N_{\mathrm{f}}+1}^{k} \widehat{\boldsymbol{\zeta}}_\ell, \tag{11}$$

where $N_{\mathrm{f}} > 0$ is an integer used to exclude those intermediate estimates that are too far away from the optimal tuning parameter values to be included in computing the final estimate. Equation (11) is a well-known approach to increase stability of the algorithm while maintaining competitive convergence rates to the optimal vector $\boldsymbol{\zeta}^*$ of the tuning parameters (Maryak, 1997). This is true even if the averaging would introduce a small amount of bias, since the improved stability usually results in a more reliable optimization procedure (Kushner and Yin, 2003).

2.4 Practical Guidelines

The implementation of the SPSA optimization algorithm requires proper selection of several quantities that could have an impact on its efficiency and stability. To this end, some practical guidelines have been provided in the literature (Spall, 1998, 2003), which are described in the following several parts.

2.4.1 Choice of the score function

Selection of the score function Q used in (10) depends on the goal of the optimization problem. In SPC, the ultimate goal is to minimize the $\mathrm{ARL}_{\mathrm{OC}}$. In such cases, an appropriate choice of Q is

$$\overline{Q}[\boldsymbol{\zeta}, h(\boldsymbol{\zeta})] = r^{-1} \sum_{l=1}^{r} \mathrm{RL}_{1l}[\boldsymbol{\zeta}, h(\boldsymbol{\zeta})], \tag{12}$$

where $\mathrm{RL}_{1l}[\boldsymbol{\zeta}, h(\boldsymbol{\zeta})]$ is the l-th independently simulated RL value under the OC scenario using the tuning parameter vector $\boldsymbol{\zeta}$ and the control limit $h(\boldsymbol{\zeta})$. Other choices are also

possible. For instance, if one is interested in minimizing the MRL_{OC}, then we can replace the quantity in Equation (12) by the following median:

$$\overline{Q}[\boldsymbol{\zeta}, h(\boldsymbol{\zeta})] = \text{median}\{\text{RL}_{11}[\boldsymbol{\zeta}, h(\boldsymbol{\zeta})], \ldots, \text{RL}_{1r}[\boldsymbol{\zeta}, h(\boldsymbol{\zeta})]\}.$$

Regarding the value of r, based on the numerical results in Section 4.3, it appears that the results are reasonable good when $r \geq 100$ to obtain an improvement in algorithm convergence without sacrificing much computational efficiency.

2.4.2 Gain sequences

The gain sequences in the algorithm are commonly defined as $a_k = a/(k + A + 1)^{\alpha}$ and $c_k = c/(k + 1)^{\beta}$, where α and β are pre-specified to be 0.602 and 0.101, respectively (Spall, 2003). These gain sequences would result in a slow gain decay and ensure the convergence of $\widehat{\boldsymbol{\zeta}}_k$ to $\boldsymbol{\zeta}^*$ as $k \to \infty$ under some quite general assumptions, as proved by Spall (1992). Although there are faster convergence options available, slower rates are usually recommended since they can provide a more thorough exploration of the set $\mathcal{Z}$ and lead to a more stable algorithm in practice.

The constants a, A, and c in the above gain sequences need to be chosen carefully to guarantee a good convergence of the procedure. To this end, a preliminary adaptive step is usually employed to find proper values of these constants. More specifically, the constant c can be approximately set to be the standard deviation $\sigma_{\widehat{\boldsymbol{\zeta}}_0}$ of the OC RL calculated at the initial value $\widehat{\boldsymbol{\zeta}}_0$ (Spall, 1998). As discussed in Section 2.4.1, r replicated evaluations of the OC RL are often used to reduce variability of the gradient estimate, from which the standard deviation of the OC RL can be computed by $\widehat{\sigma}_{\widehat{\boldsymbol{\zeta}}_0} = \widehat{\sigma}_{\text{RL}_1}/\sqrt{r}$, where $\widehat{\sigma}_{\text{RL}_1}$ is the standard deviation of the r simulated values of the OC RL. In practice, it can happen that the initial iterations of the algorithm move the tuning parameters too far away from the optimal values. Numerical studies show that setting $c = \min\{\widehat{\sigma}_{\widehat{\boldsymbol{\zeta}}_0}, 0.1\}$ can avoid excessive perturbation of the tuning parameters in the early iterations. According to Spall (1998),

the constant A can be set to be 0.1 times the expected number of function evaluations. For example, the expected number of evaluations used in this paper is 150, resulting in $A = 0.1 \times 150 = 15$. Once c and A are selected, Spall (1998) recommends selecting a to be the expected magnitude change in $\widehat{\boldsymbol{\zeta}}_k$ during the first few iterations. Specifically,

$$a = s \cdot (A+1)^{\alpha} / \overline{G},$$

where s is the initial step size and $\overline{G} = \frac{1}{d}\sum_{j=1}^{d}\sum_{l=1}^{n_c} \widehat{\boldsymbol{g}}_{jl}(\widehat{\boldsymbol{\zeta}}_0)/n_c$ is a preliminary estimate of the average value of the gradient in $\widehat{\boldsymbol{\zeta}}_0$ based on n_c simulated RLs. For instance, a reasonable initial step size s for an EWMA chart could be 0.2, and setting $n_c = 20$ is found to be appropriate to estimate the gradient at the beginning of the algorithm.

2.4.3 Convergence criteria

A reasonable stopping rule of the algorithm is $|\,\mathbb{E}[\widehat{\boldsymbol{g}}_{jk}(\widehat{\boldsymbol{z}}_k)]| \leq \nu$, for all $j = 1, \ldots, d$, where ν is a pre-specified accuracy level. From the score expression (10), it is possible to obtain an estimate of the gradient's variance at each iteration. Rewrite the gradient estimate in (10) as $\bar{\boldsymbol{g}}_k(\widehat{\boldsymbol{\zeta}}_k) = G_k \boldsymbol{D}_k$, where

$$G_k = \frac{\overline{Q}\left[\widehat{\boldsymbol{\zeta}}_k^{+}, h(\widehat{\boldsymbol{\zeta}}_k^{+})\right] - \overline{Q}\left[\widehat{\boldsymbol{\zeta}}_k^{-}, h(\widehat{\boldsymbol{\zeta}}_k^{-})\right]}{2c_k},$$
$$D_{jk} = \Delta_{jk}^{-1} \quad \text{for } j = 1, \ldots, d.$$

Then, the variance of the jth element of the gradient estimate has the expression

$$\begin{aligned}
\mathbb{V}[\bar{g}_{jk}(\widehat{\boldsymbol{\zeta}}_k)] &= \mathbb{E}[G_k^2 D_{jk}^2] - \left(\mathbb{E}[G_k D_{jk}]\right)^2 \\
&= \mathbb{V}[G_k]\,\mathbb{V}[D_{jk}] + \mathbb{V}[G_k]\,\mathbb{E}[D_{jk}]^2 + \mathbb{E}[G_k]^2\,\mathbb{V}[D_{jk}] \\
&= \mathbb{V}[G_k] + \mathbb{E}[G_k]^2,
\end{aligned}$$

where the second equality follows by the independence of G_k and D_k, and the third equality holds since D_{jk} is the symmetric Rademacher random variable defined in (9), for which $\mathbb{V}[D_{jk}] = 1$ and $\mathbb{E}[D_{jk}] = 0$. At the optimal tuning parameter values, $\mathbb{E}[G_k]^2 = 0$ and thus calculating the variance of the gradient only requires estimating $\mathbb{V}[G_k]$. Then, a stopping rule based on the estimated variance can be introduced using an approach similar to the one in Capizzi and Masarotto (2016).

Let $Q'(\boldsymbol{\zeta}) = \partial Q[\boldsymbol{\zeta}, h(\boldsymbol{\zeta})]/\partial \boldsymbol{\zeta}$ and assume that $Q''(\boldsymbol{\zeta}) = \partial Q'(\boldsymbol{\zeta})/\partial \boldsymbol{\zeta}^\top$ exists. Then, based on the asymptotic distribution of the Polyak-Ruppert averaging scheme (Dippon and Renz, 1997; Maryak, 1997), we have

$$k^{1/3}\left[Q'(\bar{\boldsymbol{\zeta}}_k) - Q'(\boldsymbol{\zeta}^* - \mu)\right] \sim N_d(\mathbf{0}, Q''(\boldsymbol{\zeta}^* - \mu)\Sigma Q''(\boldsymbol{\zeta}^* - \mu)^\top),$$

where $\boldsymbol{\mu}$ and Σ are quantities whose exact expressions depend on both the gain constants of the SPSA algorithm and the characteristics of the objective function. Since the asymptotic bias goes to zero when $k \to \infty$, a stopping time $\overline{N}_s \in \mathbb{N}$ based on testing whether the gradient is zero can be written as

$$\overline{N}_s = \inf\left\{k > N_\mathrm{m} + N_\mathrm{f} : k \geq \left(\frac{z}{\nu}\right)^2 \max_{j=1,\ldots,p} \frac{1}{N - N_\mathrm{f}} \sum_{\ell=N_\mathrm{f}+1}^{k} \bar{\boldsymbol{g}}_{j\ell}(\widehat{\boldsymbol{\zeta}}_k)^2\right\}, \tag{13}$$

where z is the $[(1-\nu)/2]$-th quantile of the standard normal distribution and ν is a pre-specified small value. In the above definition of $\overline{N}_s$, the lower bound $N_\mathrm{m} + N_\mathrm{f}$ is specified to avoid an early ending of the algorithm.

In some cases, the variance Q could be very large, leading to extremely large values of $\overline{N}_s$. For example, this occurs when minimizing ARL_OC of a univariate EWMA control chart for a mean shift of $\delta = 0.25$. In such cases, it has been observed that $\widehat{\boldsymbol{\zeta}}_k$ oscillates around the optimal value, instead of converging slowly to the optimum. Therefore, a stopping rule based on the gradient variance such as (13) may lead to an overly conservative termination of the algorithm in this case. However, these are also the cases when the running averaging idea described in (11) can be the most helpful, since the estimates $\widehat{\boldsymbol{\zeta}}_k$ oscillate around the

true optimal value and their average could provide a good estimate of the optimal value. See Maryak (1997) for a detailed discussion. Therefore, in such cases, it is more appropriate to use the following stopping rule:

$$\overline{N}_a = \inf\left\{k > N_{\mathrm{m}} + N_{\mathrm{f}} : \|\overline{\boldsymbol{\zeta}}_k - \overline{\boldsymbol{\zeta}}_{k-1}\| < \varepsilon\right\}, \tag{14}$$

where ε is a pre-specified small number (e.g., $\varepsilon = 10^{-5}$). By combining (13) and (14), the following stopping rule is suggested in this paper:

$$\overline{N} = \min\left\{\overline{N}_s, \overline{N}_a\right\}, \tag{15}$$

where the suggested values for $\nu, z, N_{\mathrm{f}}, N_{\mathrm{m}}$, and ε can be found in Table 1.

2.4.4 Constrained optimization

The proposed SPSA approach requires the calculation of the control limits $h(\widehat{\boldsymbol{\zeta}}^{+})$ and $h(\widehat{\boldsymbol{\zeta}}^{-})$ in each iteration. To this end, the SA algorithm described in Capizzi and Masarotto (2016) has been used here, albeit with a lower precision than that recommended in that paper. The SA algorithm is illustrated in Appendix A, and readers are referred to Capizzi and Masarotto (2016) for a further discussion on the quantities involved. A reduced precision is justified in the current research problem by the fact that $\mathrm{ARL}_{\mathrm{IC}}$ is a monotonic function of the control limits and thus the optimization is computationally easier than that in cases with non-monotonic functions. Based on our numerical experience, 100 iterations of the first stage are usually sufficient to obtain a rough estimate of the optimal Robbins-Monro gain, and a total of 100 iterations of the Polyak-Ruppert averaging are typically enough to obtain a reasonable estimate of the control limits. For these reasons, the number of iterations N_{fixed} in the adaptive stage of the algorithm, as well as the maximum number of iterations in the algorithm, are both set to be 100. In addition, to ensure that the control limits are close enough to the solution, the first half of the iterations in the Polyak-Ruppert averaging is discarded. The other parameters of the SA algorithm are set to be the default

values suggested in Capizzi and Masarotto (2016). As illustrated in Capizzi and Masarotto (2009), it is possible to modify the SA algorithm in order to enforce a constraint on the quantiles of the IC RL distribution, instead of ARL_{IC}. This modification is discussed in Appendix A.1.

After the SPSA algorithm is terminated and the estimated optimal tuning parameters $\boldsymbol{\zeta}^*$ are obtained, a more accurate estimate of the control limit can be obtained by reapplying the control limit optimization with an increased precision. The entire SPSA optimization procedure is described in Algorithm 1, and the recommended values for the parameters used in the proposed algorithm are given in Table 1.

Algorithm 1 Proposed SPSA Algorithm for Constrained Optimization

Input: $\alpha, \beta, a, A, c, r, N_{\text{m}}, N_{\text{f}}$ (see Table 1).

for $k = 0, 1, 2, \ldots$ **do**
  $c_k \leftarrow c/(1+k)^{\beta}$
  $\Delta_{jk} \overset{\text{iid}}{\sim}$ Rademacher(0.5) for all $j = 1, \ldots, d$
  $\widehat{\boldsymbol{\zeta}}_k^{+} \leftarrow \Psi(\widehat{\boldsymbol{\zeta}}_k + c_k \boldsymbol{\Delta}_k)$
  $\widehat{\boldsymbol{\zeta}}_k^{-} \leftarrow \Psi(\widehat{\boldsymbol{\zeta}}_k - c_k \boldsymbol{\Delta}_k)$
  Calculate $h(\widehat{\boldsymbol{\zeta}}_k^{+})$ and $h(\widehat{\boldsymbol{\zeta}}_k^{-})$ using the SA algorithm (Appendix A)
  Calculate $\bar{\boldsymbol{g}}_k(\widehat{\boldsymbol{\zeta}}_k)$ using (10)
  $a_k \leftarrow a/(1+A+k)^{\alpha}$
  $\widehat{\boldsymbol{\zeta}}_{k+1} \leftarrow \Psi\big(\widehat{\boldsymbol{\zeta}}_k - a_k \bar{\boldsymbol{g}}_k(\widehat{\boldsymbol{\zeta}}_k)\big)$
  Calculate $\overline{N}$ using (13), (14), and (15)
  if $k > \overline{N}$ **then**
    $\overline{N} \leftarrow k$
    break
  end if
end for

Output: $\bar{\boldsymbol{\zeta}} = \frac{1}{\overline{N} - N_{\text{f}}} \sum_{k=N_{\text{f}}+1}^{\overline{N}} \widehat{\boldsymbol{\zeta}}_k$

3 Some Numerical Results

In this section, the optimization of ARL_{OC} for the following two control charts is discussed.

1. The two-sided EWMA chart **with constant control limits** that is designed for detecting mean changes of a process (Roberts, 1959), under the assumption that the

Table 1: Recommended parameter values for implementing the proposed SPSA algorithm.

SPSA algorithm						
$r = 100$						see Section 2.4.1
$\alpha = 0.601$	$\beta = 0.101$	$n_c = 20$	a	A	c	see Section 2.4.2
$\nu = 0.05$	$z = 3$	$N_{\text{m}} = 300$	$N_{\text{f}} = 100$	$\varepsilon = 10^{-5}$		see Section 2.4.3
SA algorithm (Appendix A)						
$N_{\text{fixed}} = 100$	$N_{\text{max}} = 100$					see Section 2.4.4

IC process distribution is the standard Normal distribution. The initial value of its smoothing constant λ is sampled from the Beta(10,10) distribution. The estimated optimal parameter value for detecting a given mean change is compared with the value obtained **using the `R` package `spc`. This package provides functions for computing the IC and OC ARL values of an EWMA control chart, using the Gauss-Legendre quadrature. The optimal value of the smoothing constant λ is then determined using the `optimize()` function, which uses a combination of golden section search and parabolic interpolation.** The value of ARL_0 can change among $\{50, 100, 250, 370\}$, which are commonly used in the literature as discussed in Crowder (1989). The considered mean shifts are equally spaced in the range $[0.25, 4.0]$ with the step size of 0.25.

2. The R-SADA chart, **which is** designed for detecting mean changes in a partially observed multivariate data stream (Xian et al., 2019). The control chart is based on a two-step procedure at each time t. In the first step, the observed data are used to construct an augmented vector that contains information on all the observed and unobserved data streams. In the second step, the augmented vector is used to calculate the charting statistic using a formulation similar to the multivariate CUSUM charting statistic proposed by Crosier (1988). The IC data streams are assumed to be independent and identically distributed at different observation times and the IC process distribution is assumed to be standard Normal. In this case, it is assumed that there are $p = 100$ independent data streams, of which $q = 20$ can be observed at every observation time. After the process becomes OC, each of $n = 10$ randomly

selected data streams has a mean shift of size $\delta \in \{0.25, 0.5, 1, 2\}$. In this chart, the tuning parameters are the allowance k of the CUSUM chart and the minimum shift size $\mu_{\min}$ to be detected by the chart. Without an analytic solution for the optimal tuning parameter values, the results of the SPSA procedure are compared with the recommended values $k = 0.3$ and $\mu_{\min} = 1.5$ by Xian et al. (2019) in terms of ARL_{OC}, since the related chart is shown to be robust to a wide variety of shift sizes. These recommended tuning parameter values are also used as the initial estimates of the optimal tuning parameters in the optimization procedure.

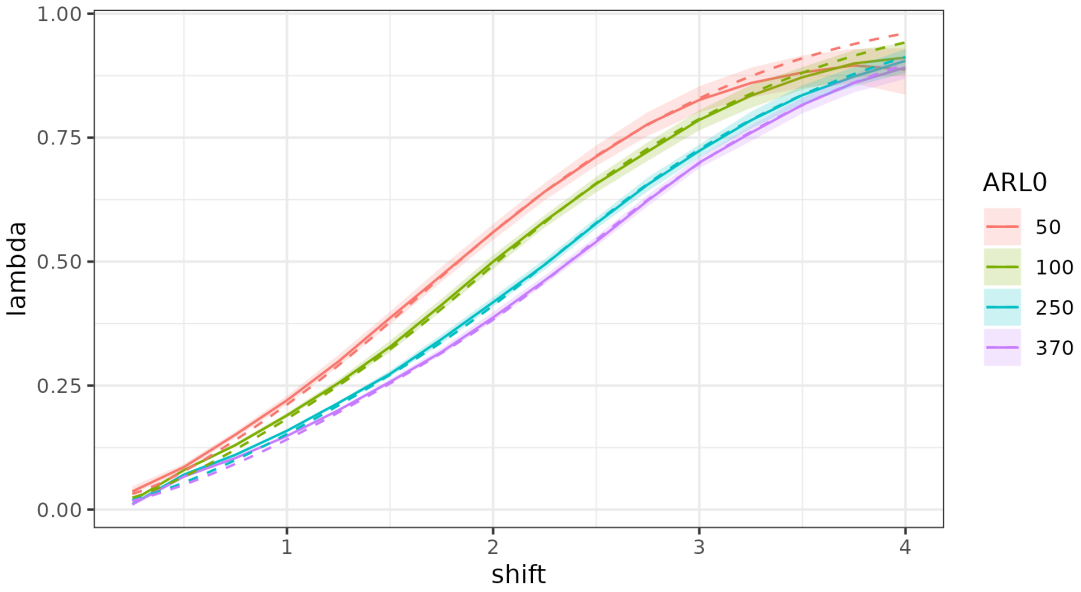

Figure 1: Estimated optimal values of λ for the EWMA chart obtained from 200 independently replicated optimizations. The solid curves denote the medians, and the shaded intervals indicate the 10th and 90th percentiles of the estimated optimal values. The dashed curves represent the optimal values obtained by the Gauss-Legendre quadrature procedure via the `R` package `spc`.

The proposed SPSA algorithm was run 200 times on both control charts, using the initial expected step size of 0.2 for the EWMA chart and 0.1 for the R-SADA chart. The estimated optimal smoothing parameters for the EWMA chart for several selected values of ARL_0 are displayed in Figure 1. The results of the SPSA method appears to be comparable to those obtained by the Gauss-Legendre quadrature procedure via the `R` package `spc`. Furthermore, it can be observed that variation of the estimated optimal tuning parameter

Table 2: Estimated optimal values of λ of the EWMA chart for detecting mean shifts of size δ in two different settings when ARL and MRL objective functions are used. The results shown in the table are medians based on 200 independent optimizations.

δ	$\min \mathrm{ARL}_{\mathrm{OC}}$ s.t. $\mathrm{ARL}_{\mathrm{IC}} = 100$	$\min \mathrm{MRL}_{\mathrm{OC}}$ s.t. $\mathrm{MRL}_{\mathrm{IC}} = 100$
0.25	0.0187	0.0138
0.50	0.0798	0.0746
0.75	0.1301	0.1210
1.00	0.1893	0.1762
1.25	0.2552	0.2384
1.50	0.3275	0.3051
1.75	0.4133	0.3796
2.00	0.4982	0.4586
2.25	0.5817	0.5450
2.50	0.6549	0.6387
2.75	0.7216	0.7016
3.00	0.7842	0.7609
3.25	0.8365	0.8116
3.50	0.8718	0.8510
3.75	0.9004	0.8731
4.00	0.9117	0.8843

values is generally small, as indicated by the relatively small differences between the 10th and 90th percentiles of the estimated optimal values, which confirms the robustness of the algorithm to the randomly selected starting points. For a combination of relatively large δ and small ARL_0, the plot shows that the estimated optimal value is a little different from the optimal solution obtained by the `R` package `spc`. This would not be a significant concern since the control chart can promptly detect the shift in such a case, even with a slightly suboptimal smoothing parameter value.

As previously discussed, the optimization problem (5) allows to use alternative objective functions other than $\mathrm{ARL}_{\mathrm{OC}}$ and $\mathrm{ARL}_{\mathrm{IC}}$. For instance, Knoth (2015) considers the calibration of an EWMA chart using a pre-specified value of $\mathrm{MRL}_{\mathrm{IC}}$, and presents some results of $\mathrm{MRL}_{\mathrm{OC}}$ for detecting certain mean shifts when the smoothing parameter λ takes a value in a given set. The related calculation is based on numerical approximations of the run length function, and is tailored for the EWMA charting statistic. Modifications of the

Table 3: Summary statistics of the estimated optimal parameter values of the R-SADA chart by the proposed SPSA approach. The results are based on 200 independent optimizations.

	k						$\mu_{\min}$					
δ	Mean	SD	min	q_{10}	q_{90}	max	Mean	SD	min	q_{10}	q_{90}	max
0.25	0.255	0.009	0.221	0.244	0.265	0.271	0.850	0.191	0.359	0.604	1.073	1.379
0.50	0.248	0.007	0.233	0.240	0.257	0.262	1.353	0.077	1.181	1.249	1.446	1.510
1.00	0.267	0.031	0.175	0.221	0.304	0.318	1.265	0.360	0.284	0.790	1.745	2.086
2.00	0.163	0.065	0.093	0.117	0.290	0.318	2.062	0.626	0.178	0.974	2.457	2.626

proposed optimization procedure to handle such cases are discussed in Section 2.4.1 and Section 2.4.4, and the results when MRL_0 is fixed at 100 are presented in Table 2.

The summary statistics of the estimated optimal tuning parameter values for the R-SADA chart are presented in Table 3 when ARL_{IC} is set to be 370. From the table, it can be seen that the estimated optimal value of $\mu_{\min}$ increases with δ, which is consistent with its interpretation given by Xian et al. (2019) as being the smallest expected mean shift. The comparison between the optimized and non-optimized OC performance of the R-SADA chart is shown in Figure 2, where ARL_{OC} is estimated based on 10,000 replicated simulations in both cases when the estimated optimal parameter values by the proposed method and the suggested parameter values by Xian et al. (2019) are used in each simulation. The plot shows a superior performance of the optimized results in comparison to the non-optimized results, especially when the shift size is small.

4 Comparisons with Traditional Approaches

In the literature, traditional approaches to optimize the tuning parameters of a control chart involve estimation of the expected values in (5) using a large number of replicated simulations. Then, under the assumption that the related expected values are deterministic, the constrained optimization is carried out by using either a grid search or a numerical searching algorithm. In this section, a simulation experiment is carried out to compare the proposed SPSA algorithm with some traditional approaches, as described below.

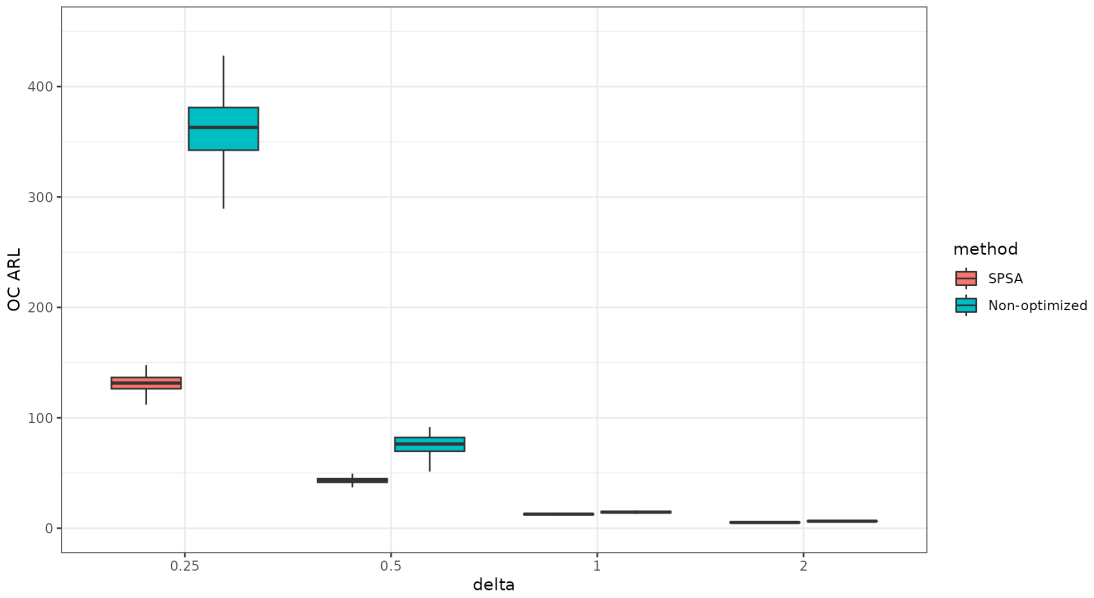

Figure 2: Comparison of the calculated ARL_{OC} values when the estimated optimal parameter values by the proposed method and the suggested parameter values by Xian et al. (2019) are used. Each ARL_{OC} value is calculated based on 10,000 replicated simulations.

1. The classical univariate CUSUM control chart for detecting upward process mean shift is considered (Page, 1954). In this case, the true optimal allowance value is known for detecting a given mean shift. The proposed SPSA method is compared with the following two alternative methods:

   (a) The grid search algorithm described in Qiu (2008), which is briefly summarized below. First, the algorithm specifies a range $[l, u]$ for the tuning parameter. Then, the interval is divided into m equally spaced subintervals. The objective function is then evaluated at each endpoint using 10,000 replicated simulations. The endpoints adjacent to the one with the smallest value of the objective function are used to define a new range for the tuning parameter. This iterative process continues until the convergence criterion $|\widehat{\boldsymbol{\zeta}}^{(k)} - \widehat{\boldsymbol{\zeta}}^{(k-1)}| < \varepsilon'$ is met, where k is the index of the iterations. In this numerical study, we choose $m = 3$, and the control limit at each endpoint is obtained by using the SA algorithm in the recommended setting discussed in Capizzi and Masarotto (2016), which has been shown to be more efficient than the classical bisection search algorithm.

(b) The Nelder-Mead optimization procedure (Nelder and Mead, 1965), which is a well-known optimization technique for handling nonlinear objective functions. This algorithm is one of the default optimizers in the `NLopt` optimization library of the Julia wrapper (Johnson, 2023). Similarly to the grid search, the objective function is evaluated each time based on 10,000 replicated simulations, and the control limit is obtained by using the SA algorithm. The algorithm is run with the same convergence criterion as that of the grid search algorithm, for a maximum of 1,000 iterations.

2. The scalability of the proposed SPSA method is studied when the chart tuning parameters are d-dimensional. In this case, the method is compared with the multivariate generalization of the two competing methods discussed above. The multivariate generalization of the grid search method starts in defining an initial d-dimensional rectangle $[\boldsymbol{l}, \boldsymbol{u}]$ from which the optimal tuning parameters are searched. The interval in each dimension is divided into m segments and the algorithm proceeds analogously to the univariate case by considering the $(m+1)^d$ lattice points defined by the segments. The procedure continues until the convergence criterion $\|\widehat{\boldsymbol{\zeta}}^{(k+1)} - \widehat{\boldsymbol{\zeta}}^{(k)}\| < \varepsilon'$ is met. Again, we use $m = 3$ in the simulation study. The Nelder-Mead algorithm uses the same convergence criterion as that of the multivariate grid search algorithm, for a maximum of 1,000 iterations.

4.1 Accuracy of the estimated optimal tuning parameters

The efficiency of the proposed algorithm is compared with the two competitors when optimizing the allowance constant of a standard CUSUM chart for detecting process mean shifts when $\text{ARL}_{\text{IC}} = 370$, the IC process distribution is $N(0, 1)$, and the objective function is ARL_{OC}. In such cases, the true optimal value of the allowance is $k = \delta/2$, where δ is a given mean shift size. The simulation study is carried out in both cases when a) the IC mean and variance are assumed known, and b) they are assumed unknown and estimated

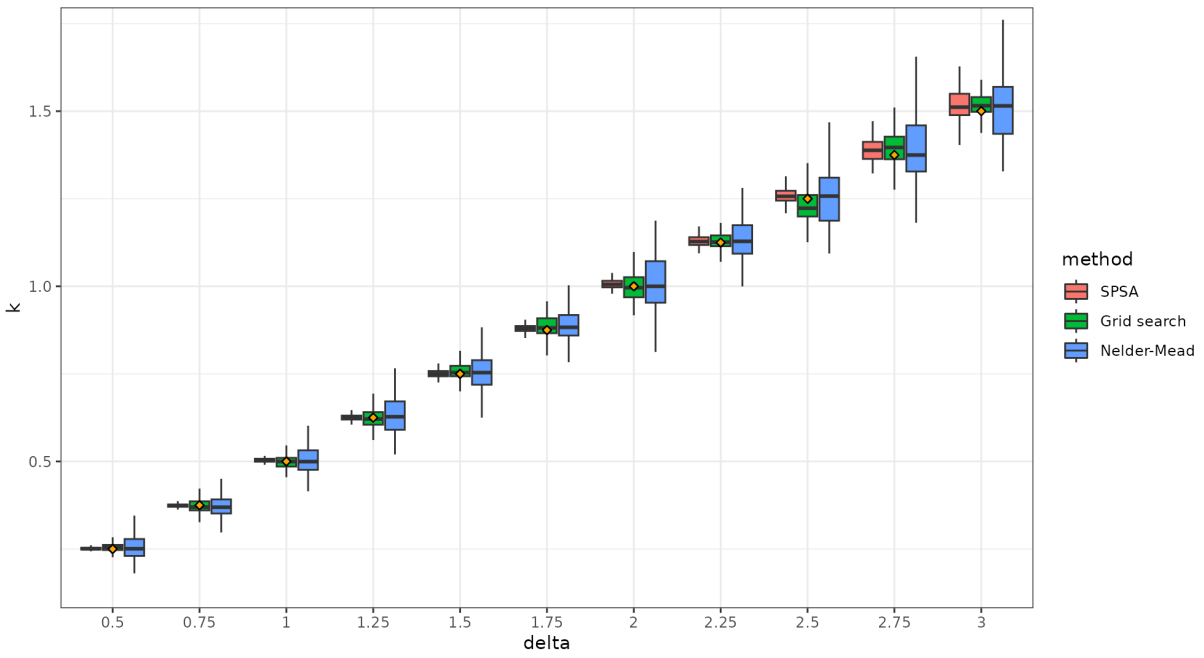

Figure 3: Comparison of accuracy of the estimated optimal tuning parameter between the proposed SPSA approach and the two traditional approaches, when the true optimal parameter values (orange dots in the plot) are known.

from an initial IC sample with 500 observations. The comparison among the three methods is about the accuracy of their estimated optimal allowance values and their running times as well. All algorithms are run 200 times using the same value of tolerance $\varepsilon = \varepsilon' = 10^{-5}$. The parameter ν used in (13) is set to be $\nu = 0.05$. In the grid search algorithm, the initial interval for searching the optimal allowance is set to be $[l, u] = [0, 4]$, and the initial value of k is set to be 1 in both the SPSA and the Nelder-Mead algorithms. When the IC parameters are being estimated, the IC and OC run lengths are obtained via **parametric** bootstrap.

The estimated optimal tuning parameter values by the three optimization methods are displayed in Figure 3. For all considered shift sizes, optimization by SPSA yields a substantially more accurate estimate of the optimal allowance in comparison with the two competing methods, since the latter are not designed specifically for optimizing random functions. More specifically, both the bias and variability of the estimated tuning parameter by the SPSA algorithm seem smaller than those by the two competing methods in most cases considered. Figure 4 shows the total running times of the three algorithms. From

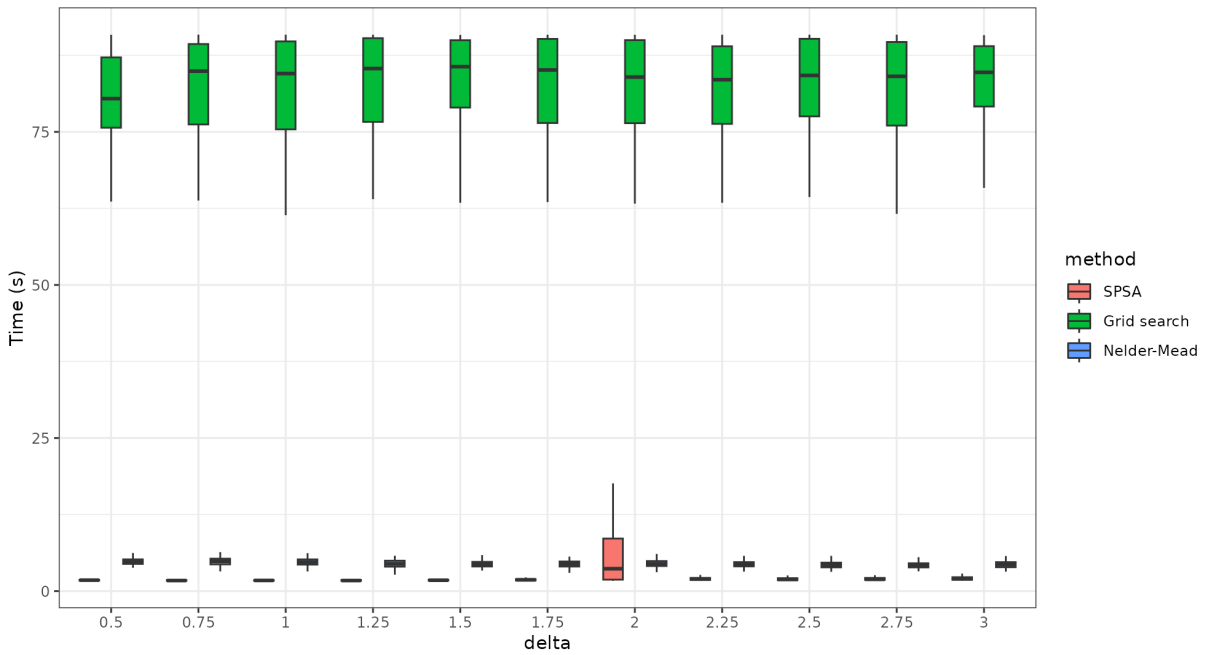

Figure 4: Computing times of the SPSA method and the two competing approaches, when the IC distribution parameters are assumed known.

the plot, it can be seen that the proposed SPSA method has a better overall performance in this regard in comparison with the two competing methods. It is only in the case when $\delta = 2$ that the SPSA algorithm takes a longer computing time than the Nelder-Mead algorithm, which is partly due to the fact that both algorithms are initialized using $k = 1$ that is precisely the true optimal value of the tuning parameter. Consequently, the SPSA algorithm initially deviates from the true optimal value and then converges to it, resulting in a longer computing time. Since the SPSA algorithm has a better performance in both the accuracy of the estimated optimal tuning parameter and computing time, we can conclude that it outperforms the two competing methods in the case considered.

When the IC distribution parameters are estimated, the resulting estimated optimal tuning parameter values are shown in Figure 5. From the plot, the estimated optimal values of the allowance are less precise in this case, compared to those when the IC parameters are assumed to be known. The precision loss is due to the randomness in the estimated IC parameters and the use of bootstrap to compute the run lengths. However, the SPSA algorithm still shows a better accuracy for small values of δ, compared to the two compet-

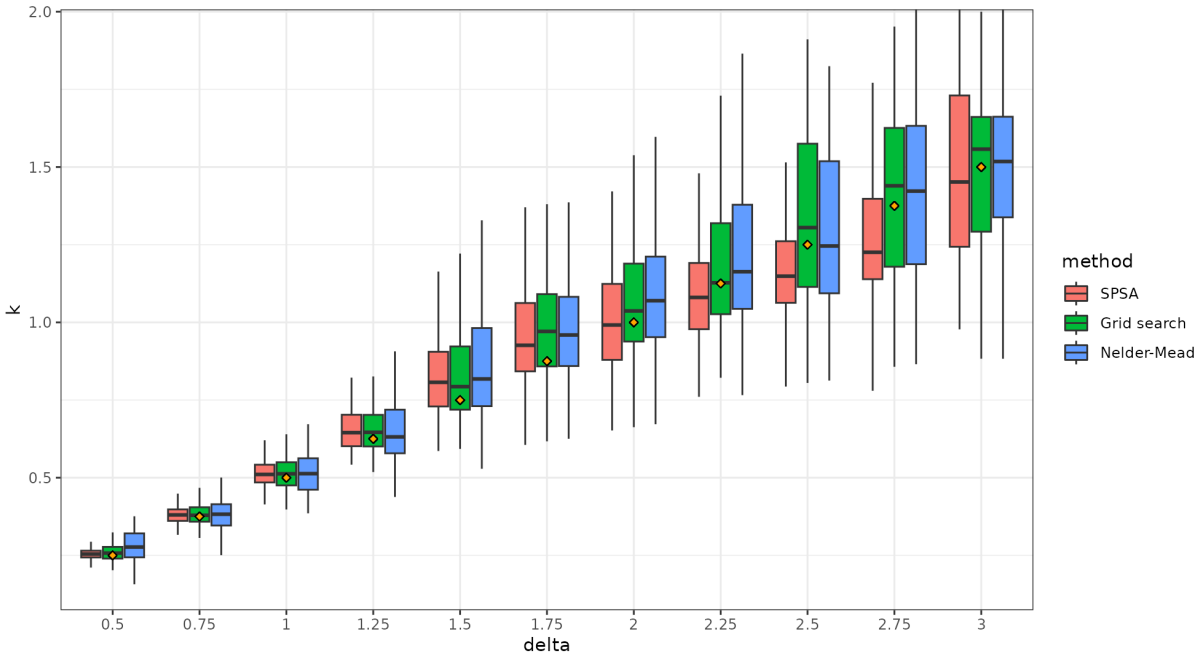

Figure 5: Comparison of the accuracy in the estimated optimal tuning parameter values by the proposed SPSA approach and the two competing approaches, when the IC distribution parameters are estimated from an initial IC sample of size 500. The orange dots in the plot indicate the true optimal tuning parameter values when the IC distribution parameters are assumed known.

ing methods. When the shift size is larger, an improvement can be seen in the reduced variability of the optimal parameter estimates, except in the case when $\delta = 3$. Figure 6 shows the computing times of the three methods. The plot shows that the proposed SPSA algorithm takes much shorter computing time than the two competing methods. **From the figure, it seems that the computing time of the grid search method in the case when $\delta = 3$ is larger than those in other cases. This might be due to the large shift size when $\delta = 3$ that would result in a large value of the allowance constant k of the related CUSUM chart. Consequently, the charting statistic C_t equals zero frequently, making estimation of the control limit computationally expensive when using Monte Carlo methods like the SA algorithm. A slight increase in computing time can also be observed for the SPSA algorithm as δ increases.**

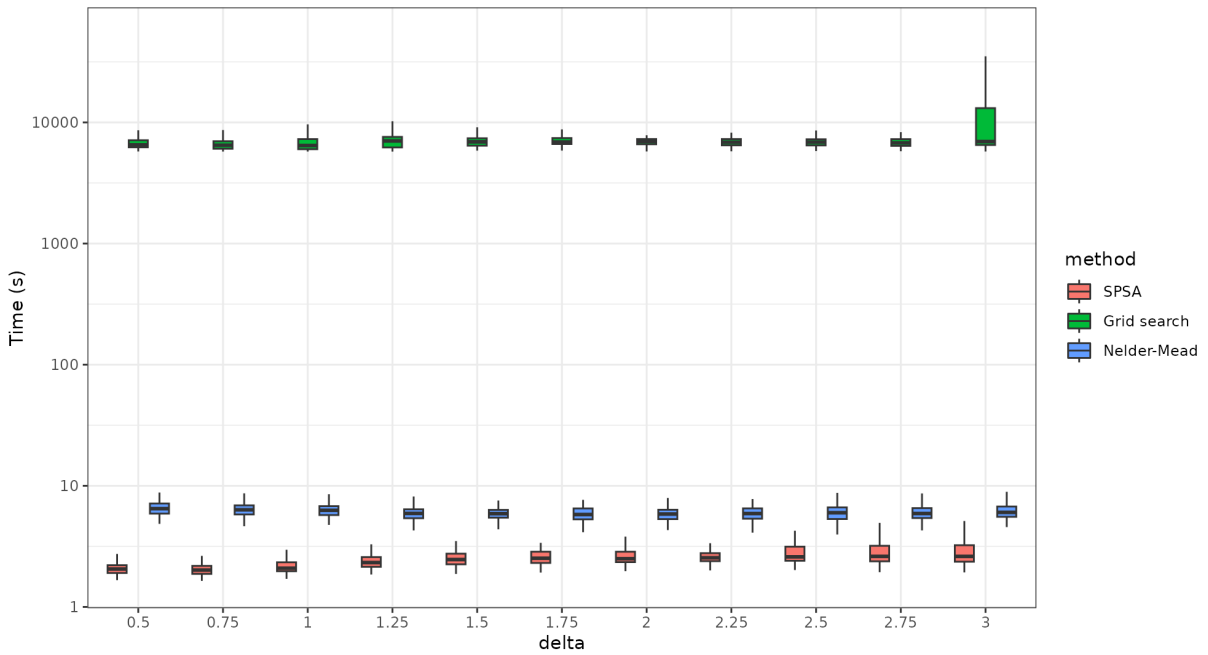

Figure 6: Computing times (in log scale) of the SPSA method and the two competing approaches, when the IC distribution parameters are estimated from an initial IC sample of size 500.

4.2 Scalability of the SPSA approach

An important feature of the proposed method is its ability to efficiently handle multidimensional optimization. In this part, we consider the Multivariate Exponentially Weighted Moving Average (MEWMA) control chart (Lowry et al., 1992) defined in (3) for monitoring the mean of a multivariate process. In the literature, it is quite common to use **the** same smoothing parameter for all quality variables when constructing the MEWMA chart for simplicity. Let us consider cases when the IC and OC distributions of the quality variables under monitoring are specified in Table 4. In these cases, the distributions of different quality variables are quite different when $d = 2$ or 3, and the use of common smoothing parameter for all variables may not be desirable. The OC settings considered are shifts in the mean of each quality variable, of size $\delta \in \{0.5, 1.0, 1.5, 2.0\}$ times the standard deviation of the related marginal distribution. **In cases when the IC marginal distribution is χ^2_1, this is achieved by changing the degrees of freedom from 1 to $1 + \delta\sqrt{2}$.** Since the related quality variables have different shapes, it is reasonable to expect that the

Table 4: IC and OC distributions for the simulated data. All variables are assumed to be independent of each other.

d	$\boldsymbol{\zeta}$	IC	OC
1	λ_1	$X_t \sim N(0,1)$	$X_t \sim N(\delta,1)$
2	(λ_1,λ_2)	$\boldsymbol{X}_t = \begin{cases} X_{1t} \sim N(0,1) \\ X_{2t} \sim \chi^2_1 \end{cases}$	$\boldsymbol{X}_t = \begin{cases} X_{1t} \sim N(\delta,1) \\ X_{2t} \sim \chi^2_{1+\delta\sqrt{2}} \end{cases}$
3	$(\lambda_1,\lambda_2,\lambda_3)$	$\boldsymbol{X}_t = \begin{cases} X_{1t} \sim N(0,1) \\ X_{2t} \sim \chi^2_1 \\ X_{3t} \sim \text{Pois}(1) \end{cases}$	$\boldsymbol{X}_t = \begin{cases} X_{1t} \sim N(\delta,1) \\ X_{2t} \sim \chi^2_{1+\delta\sqrt{2}} \\ X_{3t} \sim \text{Pois}(1+\delta) \end{cases}$

MEWMA control chart with multiple tuning parameters would perform better.

The proposed SPSA approach is compared with the multidimensional grid search algorithm and the Nelder-Mead algorithm when the number of tuning parameters is set to be $d = 1, 2, 3$, so as to study the multidimensional scalability of the algorithms. In the simulation study, ARL_0 is set to be 370, the objective function of the optimization problem is ARL_{OC}, and the tolerance criterion and algorithm constants are chosen to be the same as those in Section 4.1. The initial hypercube for the multidimensional grid search is set to $[0.01, 0.99]^d$, and the SPSA and Nelder-Mead algorithms are initialized with each smoothing parameter being 0.2. Due to the lack of an analytical solution for the true optimal tuning parameter values, the accuracy of the estimated optimal values is determined by averaging 10^6 simulated OC run lengths for each of the 200 optimizations. The optimization results are presented in Table 5. From the table, it can be seen that all algorithms obtain comparable estimates of optimal tuning parameters with similar ARL_{OC} values. However, the computing times of the three methods are dramatically different. When d increases, the SPSA algorithm is much faster than the two competing methods, especially in comparison with the multidimensional grid search algorithm.

Figure 7 shows the median and the 0.1th and 0.9th quantiles of the computing times of the three algorithms when the number of tuning parameters increases. The results clearly show that the computational cost to achieve the same accuracy in the estimated optimal tuning parameters as that in univariate cases increases very rapidly for the grid

Table 5: Comparison of optimization results using the SPSA approach, the multidimensional grid search algorithm, and Nelder-Mead optimization. All displayed values are medians calculated across 200 optimizations. Time required for the optimization is displayed in seconds.

		SPSA					Grid search					Nelder-Mead				
d	δ	λ_1	λ_2	λ_3	ARL_1	time (s)	λ_1	λ_2	λ_3	ARL_1	time (s)	λ_1	λ_2	λ_3	ARL_1	time (s)
1																
	0.5	0.071	-	-	26.784	11.698	0.053	-	-	26.426	7.544	0.053	-	-	26.443	**6.154**
	1.0	0.150	-	-	9.576	**2.885**	0.142	-	-	9.573	6.815	0.144	-	-	9.589	4.582
	1.5	0.257	-	-	5.162	**2.150**	0.256	-	-	5.167	6.287	0.256	-	-	5.175	4.379
	2.0	0.384	-	-	3.349	**2.015**	0.393	-	-	3.347	5.777	0.387	-	-	3.350	4.290
2																
	0.5	0.077	0.050	-	21.116	**17.416**	0.081	0.039	-	20.906	258.294	0.082	0.042	-	21.016	96.602
	1.0	0.150	0.065	-	8.421	**17.385**	0.164	0.051	-	8.327	257.436	0.175	0.051	-	8.351	90.400
	1.5	0.247	0.066	-	4.940	**18.640**	0.265	0.053	-	4.923	256.493	0.266	0.056	-	4.927	86.591
	2.0	0.356	0.041	-	3.414	**20.069**	0.374	0.051	-	3.418	256.984	0.361	0.050	-	3.416	82.925
3																
	0.5	0.092	0.036	0.113	16.875	**57.267**	0.115	0.045	0.140	17.443	2839.178	0.176	0.037	0.174	17.882	363.527
	1.0	0.195	0.054	0.208	6.585	**71.527**	0.218	0.058	0.218	6.700	2808.313	0.225	0.047	0.242	6.662	339.723
	1.5	0.316	0.050	0.310	3.868	**97.995**	0.356	0.064	0.311	3.945	2832.017	0.316	0.050	0.323	3.894	334.904
	2.0	0.447	0.025	0.397	2.716	**104.685**	0.486	0.054	0.407	2.731	2902.462	0.424	0.032	0.423	2.706	324.591

search algorithm. The Nelder-Mead algorithm also requires much longer computing time when d increases. As a comparison, the SPSA algorithm displays a remarkable efficiency in computing times when d increases. When $d = 3$, its computing time is much less than those of the two competing methods.

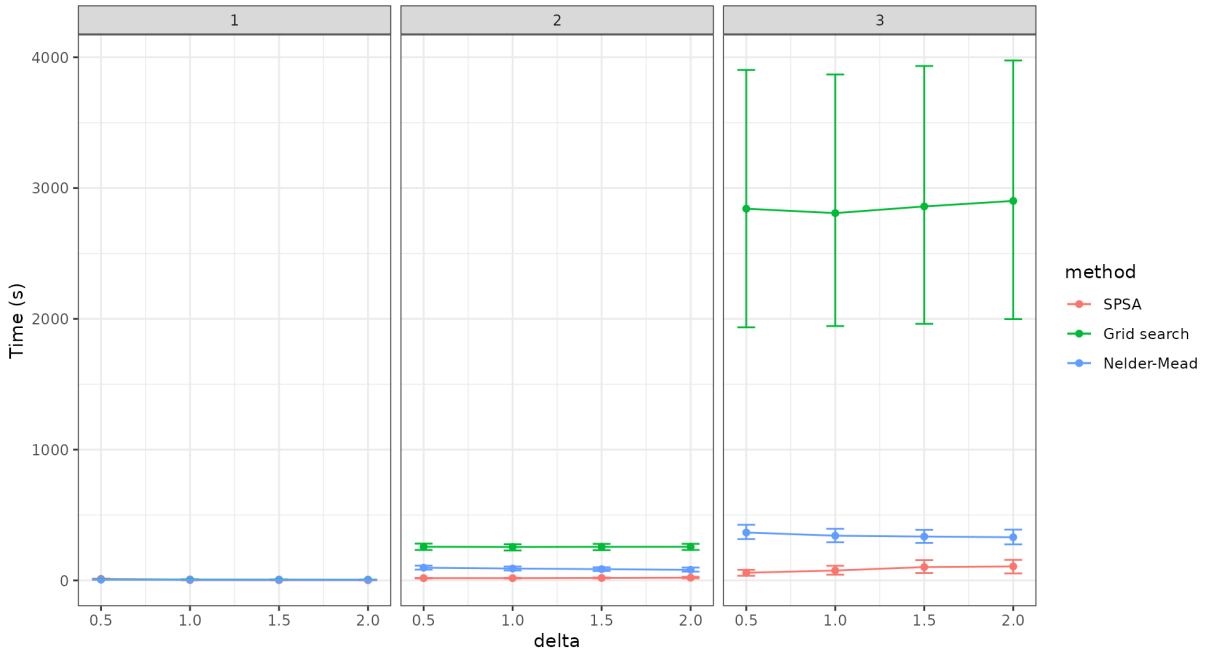

Figure 7: Median and 0.1th and 0.9th quantiles of the computing times of the SPSA algorithm and its two competing methods. The results are based on 200 independent optimizations.

4.3 On the choice of r

As mentioned in Section 2, the simple gradient estimator (8) is replaced by the gradient (10) that averages r evaluations of the OC RL in the proposed SPSA algorithm. Obviously, the computational cost would increase when r increases, with the benefit of increasing accuracy in the gradient estimates. In this part, we study the impact of r on the accuracy of the final estimate of the optimal tuning parameters and the computing time of the algorithm, using the univariate CUSUM chart for monitoring process mean under the assumption of Gaussian process observations with known IC distribution. In the simulation study, the mean shift δ can change in the set $\{0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$, and the true optimal tuning parameter is $\delta/2$. Then, 200 optimizations are performed using the SPSA algorithm, with

the initial tuning parameter value generated from the uniform distribution in the interval $[0, 3]$. The numerical study shows that choosing values of $r < 5$ often lead to a poor convergence. Figure 8 shows the median computing times of the proposed SPSA algorithm for different values of $r \in [5, 500]$. The plot shows that a small value of r would result in a larger computing time. This is because although the gradient (10) is faster to evaluate when r is small, the lower precision requiring a larger number of iterations for the algorithm to convergence. On the other hand, when r is larger than 200, it appears that the benefit of reducing the randomness in the objective function is offset by the higher computational cost. Based on these simulation results, it seems that a reasonable value of r is between 50 and 150, which balances randomness reduction and computational cost well.

Figure 9 shows the mean squared error of the estimated optimal tuning parameter in various cases considered in Figure 8. The results indicate that the accuracy of the estimate is not affected much by r when δ is small. However, for larger values of δ, using $r \geq 50$ would yield a better overall result. In conclusion, a value of r between 50 and 150 seems to provide a reasonably good balance between accuracy of the estimated optimal tuning parameter and computational time.

5 Conclusions

This paper proposes a novel approach to optimize a control chart for detecting a target OC scenario while meet constraints on some characteristics of the IC run length distribution. The proposed methodology is based on stochastic approximations and uses the SPSA algorithm (Spall, 1992) to handle optimization with a random objective function in both univariate and multivariate settings. To perform the constrained optimization, the SPSA algorithm is combined with a recently developed SA-based method for determining the control limit of the control chart (Capizzi and Masarotto, 2016). Simulation results have demonstrated the effectiveness of the proposed algorithm in comparison to some representative traditional approaches. Notably, the proposed method displays a competitive

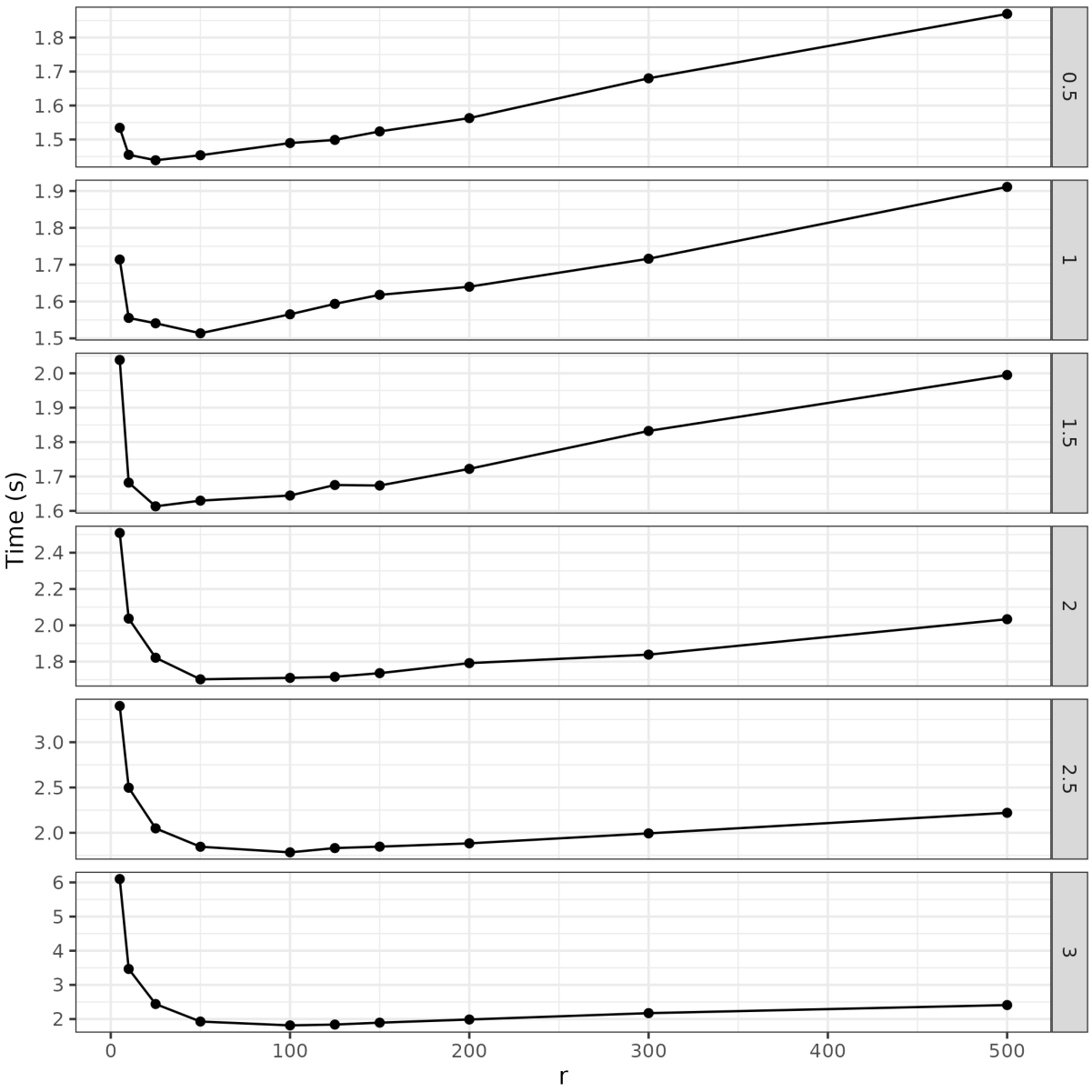

Figure 8: Median computing times of the SPSA algorithm based on 200 independent replications of the optimization procedure when r changes from 5 to 500 and the shift size δ changes among $\{0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$ (different rows).

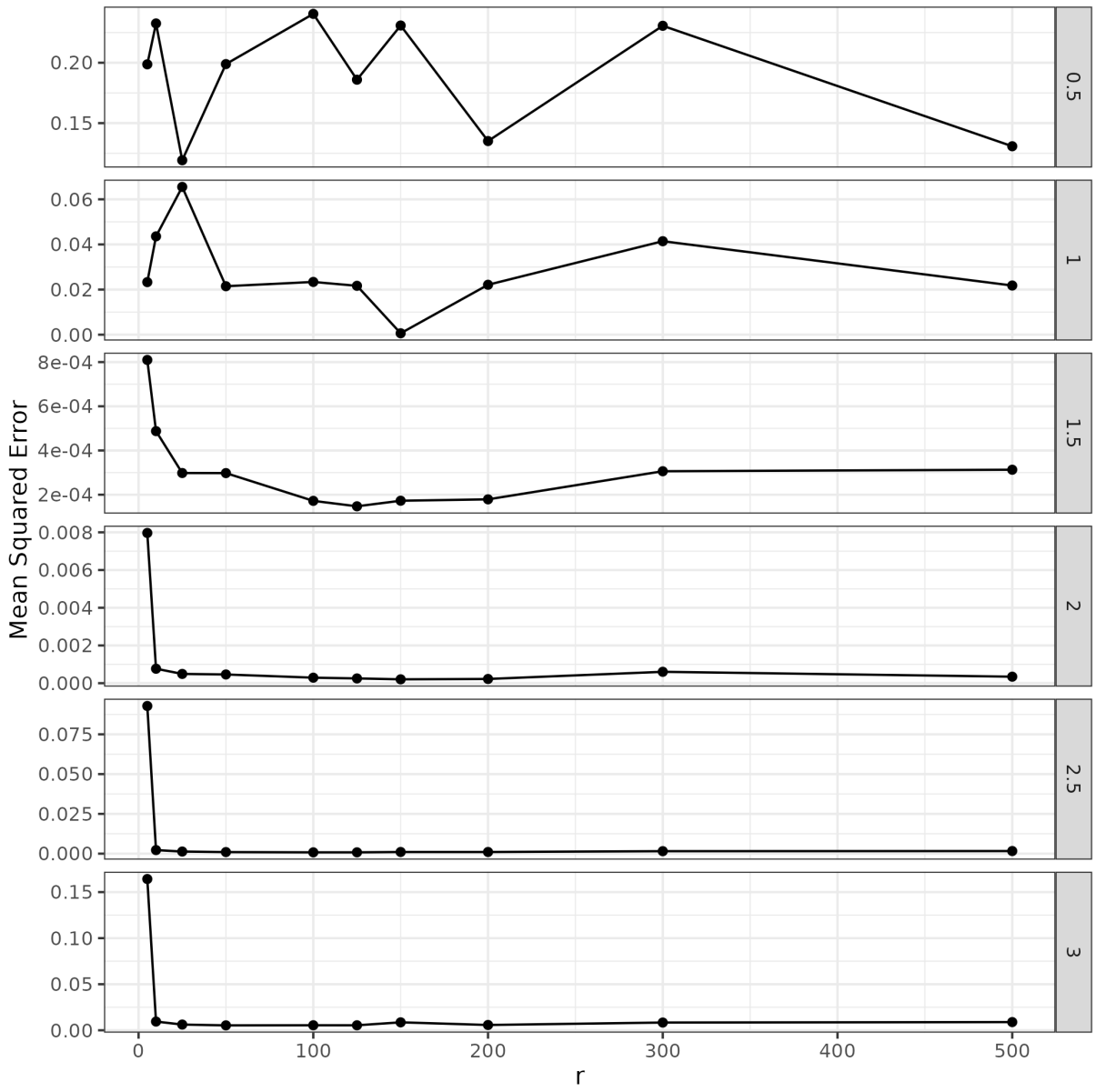

Figure 9: Mean squared error of the estimated optimal tuning parameter obtained by the SPSA algorithm in various cases considered in Figure 8. The results are based on 200 independent replications of the optimization procedure.

accuracy in comparison with the traditional methods, while **requiring** significantly less computing time, especially when the number of tuning parameters is relatively large. Although the proposed methodology is discussed mainly in cases when the OC ARL and IC ARL are used in defining the optimization problem, as highlighted in previous studies (Capizzi and Masarotto, 2009; Knoth, 2015; Capizzi and Masarotto, 2016), it can also be used when the median or other characteristics of the RL distribution are considered in the optimization. In the paper, we have discussed an adjustment of the method to optimize the median of the RL distribution. Generalizations of the method to optimize quantiles of the RL distribution other than the median are also straightforward. An efficient Julia implementation of the proposed SPSA algorithm and the SA algorithm are provided in the supplementary material.

As mentioned in Section 2, the case with multiple control charts is not considered in the present paper, although the SA algorithm as originally discussed in Capizzi and Masarotto (2016) can handle such cases. Addressing this setting requires the development of a suitable framework for quantifying the OC performance when multiple control charts are considered together, which requires much future research. One possible approach involves considering a set of OC scenarios, one for each control chart. Then, these OC scenarios can evaluated jointly, either through a weighted average of their OC performance or by sequentially optimizing each scenario by using the previous solution as a soft constraint for subsequent optimizations. However, a comprehensive analysis of such approaches falls outside the scope of this paper and is left for future research.

Data availability statement

The authors confirm that the data supporting the findings of this study are available within the article and its supplementary materials.

Competing interests statement

The authors have no relevant financial or non-financial interests to disclose

Acknowledgments

The authors are grateful to the editor and two anonymous reviewers for their many constructive comments and suggestions, which improved the quality of the paper significantly.

References

Aparisi, F. and J. C. García-Díaz (2007). Design and optimization of EWMA control charts for in-control, indifference, and out-of-control regions. *Computers & Operations Research 34*(7), 2096–2108.

Branke, J., K. Deb, K. Miettinen, and R. Slowiński (Eds.) (2008). *Multiobjective Optimization: Interactive and Evolutionary Approaches* (2008th edition ed.). Berlin ; New York: Springer.

Brook, D. and D. A. Evans (1972). An Approach to the Probability Distribution of Cusum Run Length. *Biometrika 59*(3), 539–549.

Capizzi, G. and G. Masarotto (2003). An Adaptive Exponentially Weighted Moving Average Control Chart. *Technometrics 45*(3), 199–207.

Capizzi, G. and G. Masarotto (2009). Bootstrap-based design of residual control charts. *IIE Transactions 41*(4), 275–286.

Capizzi, G. and G. Masarotto (2010). Evaluation of the run-length distribution for a combined Shewhart-EWMA control chart. *Statistics and Computing 20*(1), 23–33.

Capizzi, G. and G. Masarotto (2016). Efficient control chart calibration by simulated stochastic approximation. *IIE Transactions 48*(1), 57–65.

Chen, A. and Y. K. Chen (2007). Design of EWMA and CUSUM control charts subject to random shift sizes and quality impacts. *IIE Transactions 39*(12), 1127–1141.

Crosier, R. B. (1988). Multivariate Generalizations of Cumulative Sum Quality-Control Schemes. *Technometrics 30*(3), 291–303.

Crowder, S. V. (1987). A Simple Method for Studying Run – Length Distributions of Exponentially Weighted Moving Average Charts. *Technometrics 29*(4), 401–407.

Crowder, S. V. (1989). Design of Exponentially Weighted Moving Average Schemes. *Journal of Quality Technology 21*(3), 155–162.

Dippon, J. and J. Renz (1997). Weighted Means in Stochastic Approximation of Minima. *SIAM Journal on Control and Optimization 35*(5), 1811–1827.

Duncan, A. J. (1956). The Economic Design of X Charts Used to Maintain Current Control of a Process. *Journal of the American Statistical Association 51*(274), 228–242.

Fellner, W. H. (1990). Average Run Lengths for Cumulative Sum Schemes. *Journal of the Royal Statistical Society. Series C (Applied Statistics) 39*(3), 402–412.

Gandy, A. and J. T. Kvaløy (2013). Guaranteed Conditional Performance of Control Charts via Bootstrap Methods. *Scandinavian Journal of Statistics 40*(4), 647–668.

Hitczenko, P. and S. Kwapień (1994). On the Rademacher Series. In J. Hoffmann-Jørgensen, J. Kuelbs, and M. B. Marcus (Eds.), *Probability in Banach Spaces, 9*, Progress in Probability, Boston, MA, pp. 31–36. Birkhäuser.

Ho, C. and K. E. Case (1994). Economic Design of Control Charts: A Literature Review for 1981–1991. *Journal of Quality Technology 26*(1), 39–53.

Huang, W., L. Shu, and W. Jiang (2016). A Gradient Approach to the Optimal Design of CUSUM Charts Under Unknown Mean-Shift Sizes. *Journal of Quality Technology 48*(1), 68–83.

Huang, W., L. Shu, and W. Jiang (2018). A Gradient Approach to Efficient Design and Analysis of Multivariate EWMA Control Charts. *Journal of Statistical Computation and Simulation 88*(14), 2707–2725.

Johnson, S. G. (2023). The NLopt nonlinear-optimization package.

Jones, L. A. (2002). The statistical design of EWMA control charts with estimated parameters. *Journal of Quality Technology 34*(3), 277–288.

Knoth, S. (2015). Run length quantiles of EWMA control charts monitoring normal mean or/and variance. *International Journal of Production Research 53*(15), 4629–4647.

Knoth, S. and M. C. Morais (2015). On ARL-Unbiased Control Charts. In S. Knoth and W. Schmid (Eds.), *Frontiers in Statistical Quality Control 11*, Frontiers in Statistical Quality Control, pp. 95–117. Cham: Springer International Publishing.

Kuiper, A. and R. Goedhart (2023). Optimized control charts using indifference regions. *Quality Engineering*.

Kushner, H. and G. G. Yin (2003). *Stochastic Approximation and Recursive Algorithms and Applications* (Second ed.). New York: Springer.

Lai, T. L. (2003). Stochastic approximation: Invited paper. *The Annals of Statistics 31*(2), 391–406.

Lorenzen, T. J. and L. C. Vance (1986). The Economic Design of Control Charts: A Unified Approach. *Technometrics 28*(1), 3–10.

Lowry, C. A., W. H. Woodall, C. W. Champ, and S. E. Rigdon (1992). A Multivariate Exponentially Weighted Moving Average Control Chart. *Technometrics 34*(1), 46–53.

Lucas, J. M. and M. S. Saccucci (1990). Exponentially Weighted Moving Average Control Schemes: Properties and Enhancements. *Technometrics 32*(1), 1–12.

Mahmoud, M. A. and A. R. Zahran (2010). A Multivariate Adaptive Exponentially Weighted Moving Average Control Chart. *Communications in Statistics - Theory and Methods 39*(4), 606–625.

Maryak, J. (1997). Some guidelines for using iterate averaging in stochastic approximation. In *Proceedings of the 36th IEEE Conference on Decision and Control*, Volume 3, pp. 2287–2290 vol.3.

Nelder, J. A. and R. Mead (1965). A Simplex Method for Function Minimization. *The Computer Journal 7*(4), 308–313.

Page, E. S. (1954). Continuous Inspection Schemes. *Biometrika 41*(1/2), 100.

Pignatiello Jr, J. J., C. A. Acosta-Mejia, and B. V. Rao (1995). The performance of control charts for monitoring process dispersion. In *Proceedings of the 4th Industrial Engineering Research Conference*, pp. 320–328. Institute of Industrial Engineers Nashville, TN.

Polyak, B. and A. Juditsky (1992). Acceleration of stochastic approximation by averaging. *Siam Journal on Control and Optimization*.

Qiu, P. (2008). Distribution-free multivariate process control based on log-linear modeling. *IIE Transactions 40*(7), 664–677.

Qiu, P. (2013). *Introduction to Statistical Process Control*. CRC Press.

Rigdon, S. E. (1995a). A double-integral equation for the average run length of a multivariate exponentially weighted moving average control chart. *Statistics & Probability Letters 24*(4), 365–373.

Rigdon, S. E. (1995b). An integral equation for the in-control average run length of a multivariate exponentially weighted moving average control chart. *Journal of Statistical Computation and Simulation 52*(4), 351–365.

Robbins, H. and S. Monro (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics 22*(3), 400–407.

Roberts, S. W. (1959). Control Chart Tests Based on Geometric Moving Averages. *Technometrics 1*(3), 239–250.

Ruppert, D. (1991). Stochastic approximation. In B. K. Ghosh and P. K. Sen (Eds.), *Handbook of Sequential Analysis*, pp. 503–529. New York, NY.

Ryu, J.-H., G. Wan, and S. Kim (2010). Optimal Design of a CUSUM Chart for a Mean Shift of Unknown Size. *Journal of Quality Technology 42*(3), 311–326.

Shu, L., W. Huang, and W. Jiang (2014). A novel gradient approach for optimal design and sensitivity analysis of EWMA control charts. *Naval Research Logistics (NRL) 61*(3), 223–237.

Spall, J. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control 37*(3), 332–341.

Spall, J. (1998). Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Transactions on Aerospace and Electronic Systems 34*(3), 817–823.

Spall, J. (2003). *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control* (1. edizione ed.). Hoboken, N.J: Wiley-Interscience.

Vance, L. C. (1986). Average Run Lengths of Cumulative Sum Control Charts for Controlling Normal Means. *Journal of Quality Technology 18*(3), 189–193.

Wang, J., Z. L. Chong, and P. Qiu (2021). Optimal monitoring of Poisson data with known and unknown shifts. *Computers & Industrial Engineering 154*, 107100.

Woodall, W. H. (1985). The Statistical Design of Quality Control Charts. *Journal of the Royal Statistical Society. Series D (The Statistician) 34*(2), 155–160.

Xian, X., C. Zhang, S. Bonk, and K. Liu (2019). Online monitoring of big data streams: A rank-based sampling algorithm by data augmentation. *Journal of Quality Technology 53*(2), 135–153.

Yashchin, E. (1993). Performance of CUSUM Control Schemes for Serially Correlated Observations. *Technometrics 35*(1), 37–52.

A SA algorithm

The SA algorithm (Capizzi and Masarotto, 2016) aims at finding the control limit $h^*(\boldsymbol{\zeta})$ such that the constraint

$$K(\boldsymbol{\zeta}, h(\boldsymbol{\zeta})) = b,$$

is satisfied for a pre-determined value of b and a fixed vector of tuning constants $\boldsymbol{\zeta}$. Here, $K[\boldsymbol{\zeta}, h(\boldsymbol{\zeta})]$ is a score function based on the IC run-length distribution, such as the ARL_{IC} or the MRL_{IC}.

The algorithm is here described in a simplified version for a univariate control limit $h(\boldsymbol{\zeta})$, as this is the setting considered in the present paper. When multiple control charts are being run simultaneously, the more general formulation in Capizzi and Masarotto (2016) can be used to handle the joint optimization. Dropping the dependence of $h(\boldsymbol{\zeta})$ from $\boldsymbol{\zeta}$ for ease of notation, the algorithm is based on the Robbins-Monro (RM) recursive iteration (Robbins and Monro, 1951)

$$h_{k+1} = \max\left\{0, h_k - \frac{1}{k+1}\alpha s_{h_k}\right\}, \quad k = 0, 1, 2, \ldots, N_{\max} \tag{A.1}$$

where α is the gain of the scheme and s_{h_k} is an estimate of the gradient of K at the current iterate h_k. An in-depth discussion of the choice of score s_{h_k} is deferred to Appendix A.1.

To improve the standard RM algorithm (A.1), Capizzi and Masarotto (2016) suggest two modifications: the first modification is introduced to estimate the optimal value of α

using a preliminary adaptive stage. In the adaptive stage, the following fixed-gain recursion is employed,

$$\tilde{h}_{k+1} = \max\left\{0, \tilde{h}_k - A_{\text{fixed}} s_{h_k}\right\}, \quad k = 0, 1, \ldots, N_{\text{fixed}} - 1, \tag{A.2}$$

with $\tilde{h}_0$ being the initial control limit value and $A_{\text{fixed}} > 0$ is a scalar constant. The fixed-gain recursion in Equation (A.2) is used twofold: firstly, to provide a starting point for the iteration in (A.3), so that the initial control limit value is not too far from the solution h^*. Secondly, during the adaptive stage, the gain constant α is obtained by simulating the gradients $s_k^{\pm}$ using the control limit values $\tilde{h}_k \pm \Delta$. Then, the optimal gain matrix is defined as

$$\alpha = \frac{1}{\max\left\{\frac{1}{A_{\max}}, \min\left\{\frac{1}{A_{\min}}, d\right\}\right\}},$$

where

$$d = \frac{1}{2\Delta N_{\text{fixed}}} \sum_{k=0}^{N_{\text{fixed}}-1} (s_k^+ - s_k^-).$$

The second modification is introduced to use the iterate averaging approach, which was already discussed in Section 2.3. At the N-th recursive step, the estimate of the control limit is

$$\bar{h}_N = \frac{1}{N} \sum_{k=1}^{N} h_k,$$

where h_k is calculated using a slightly modified recursion

$$h_{k+1} = \max\left\{0, h_k - \frac{1}{(k+1)^q} \alpha s_{h_k}\right\}, \quad k = 0, 1, 2, \ldots, N_{\max}. \tag{A.3}$$

While the original implementation of the SA algorithm also makes use of a stopping rule, in this paper a truncated version is employed by setting an upper bound $N_{\max}$ for the maximum number of iterations. See Table 1 of Capizzi and Masarotto (2016) for the suggested constant values for implementing the SA algorithm.

A.1 Choice of the SA gradient

Calculation of the gradient s_{h_k} depends on the selection of a suitable score function K. Among the various options, the most commonly used score function is the $\mathrm{ARL}_{\mathrm{IC}}$. In this case, letting $\mathrm{ARL}_0 = b$, an appropriate expression for the gradient is

$$s_{h_k} = \frac{\mathrm{RL}_0[\boldsymbol{\zeta}, h_k] - \mathrm{ARL}_0}{\mathrm{ARL}_0}.$$

With this choice of gradient, the control limit h^* found by the SA algorithm satisfies the constraint $\mathbb{E}_0\{\mathrm{RL}[\boldsymbol{\zeta}, h^*]\} = \mathrm{ARL}_0$.

Another possible choice is to place a constraint on the ρ-level quantile of the IC run length. The most common selection is $\rho = 0.5$, which corresponds to constraining $\mathrm{MRL}_{\mathrm{IC}} = \mathrm{MRL}_0$. For the optimization of the ρ-level quantile of the IC run length, the following gradient can be employed

$$s_{h_k} = -\big[I(\mathrm{RL}_0(\boldsymbol{\zeta}, h_k) \leq b) - \rho\big],$$

where $I(E)$ is the indicator function of the event E. The choice of this score function ensures that the obtained control limit h^* satisfies the constraint $\mathbb{P}_0\{\mathrm{RL}[\boldsymbol{\zeta}, h^*] \leq b\} = \rho$, where $\mathbb{P}_0\{\cdot\}$ denotes the probability under the in-control process. This property follows immediately from the well-known property of indicator functions, $\mathbb{E}\{I(E)\} = \mathbb{P}\{E\}$.