An Improved Bisection-Type Algorithm for Control Chart Calibration

Daniele Zago[1*], Giovanna Capizzi[1] and Peihua Qiu[2]

[1*]Department of Statistical Sciences, University of Padua, Via Cesare Battisti 241, Padua, 35121, Italy.

[2]Department of Biostatistics, University of Florida, 2004 Mowry Rd, Gainesville, 32610, FL, USA.

*Corresponding author(s). E-mail(s): daniele.zago.1@phd.unipd.it; Contributing authors: giovanna.capizzi@unipd.it; pqiu@ufl.edu;

Abstract

Calculation of the control limits is a critical step when designing control charts in statistical process control. Traditional control chart designs require the control limits to be computed so that a characteristic of the in-control run length distribution, such as the mean or median, equals a pre-determined value. When the complexity of the in-control process distribution hinders analytical methods, Monte Carlo approaches can be used to find the appropriate control limits. Among these methods, the classical bisection searching algorithm is widely used. However, a major drawback of this method is the requirement of an initial range of values for the search. Furthermore, it is computationally very demanding when multiple control charts are used simultaneously. In this paper, we present a modified bisection searching algorithm to enhance the computational efficiency. The new method eliminates the initial specification of a range for searching. Additionally, an efficient generalization of this approach is proposed to handle the multi-chart setting. Numerical results confirm that our method offers an efficient and reliable way to compute the control limits, in comparison with the conventional bisection searching algorithm and the algorithm based on stochastic approximations. A Julia computer code implementing the proposed method is provided in the supplemental materials.

Keywords: Control limits, Monte-Carlo simulation, Bisection search, Multi-chart monitoring schemes, Stochastic approximations

1 Introduction

Control charts are a fundamental tool of statistical process control (SPC) when monitoring sequential processes with the goal of enhancing quality and detecting anomalies (Qiu, 2013). They are used to determine whether the process under monitoring has transitioned from a condition of stability (or in-control, IC) to a condition of instability (or out-of-control, OC) by the current observation time. A common measure of efficiency of a control chart is the run length (RL), which is the number of observation times required to trigger an alarm.

The design of a control chart typically involves two steps. First, specify certain nominal properties of the IC RL distribution, such as the IC average RL denoted as ARL_0 (cf., Li et al., 2014), the IC median RL denoted as MRL_0 (cf., Waldmann, 1986a,b; Gan, 1993, 1994; Graham et al., 2017; Hu et al., 2021; Qiao et al., 2022), and the quantiles of the IC RL distribution (cf., Knoth, 2015). Second, determine the values of the control limits involved in the charting statistic to meet the pre-specified nominal properties of the IC RL distribution.

The IC RL characteristics generally depend, in addition to the control limits, on the underlying IC process distribution and other tuning parameters such as smoothing and allowance constants. In practical applications, these tuning parameters are selected prior to the start of the monitoring phase by minimizing certain characteristics of the OC RL, such as the average run length given the expected OC scenario (cf., Qiu, 2008; Mahmoud and Zahran, 2010). As a result, the calibration of the control limits has to be repeated for every value of the tuning parameters considered in the minimization of the OC RL characteristics. Thus, efficient determination of the control limits can have a significant impact when applying control charts in real-world applications.

In traditional SPC problems with independent and identically distributed IC process observations, ARL_0 or other characteristics of the IC RL distribution could be computed analytically for conventional control charts, such as the Shewhart (Shewhart, 1931), EWMA (Roberts, 1959; Crowder, 1989), and CUSUM (Page, 1954; Crosier, 1986) charts. In many modern settings, however, the characteristics of interest of the IC RL distribution could be too complicated to be approximated adequately using either analytical methods or deterministic numerical methods. This complexity may arise due to autocorrelation in the process observations (Montgomery and Mastrangelo, 1991; Capizzi and Masarotto, 2008, 2009; Qiu and You, 2022) and complex IC process distributions in cases such as multi-stage or spatial processes (Jin and Shi, 1999; Huang et al., 2002; Yang and Qiu, 2020; Zhou et al., 2003) and partially-observed processes (Liu et al., 2015; Xian et al., 2018; Ye et al., 2023).

Nowadays, it is also common to combine multiple control charts for monitoring multiple process parameters or for enhancing detection power when small and large shifts are of interest (Gan, 1995; Han and Tsung, 2007; Han et al., 2007; Reynolds and Stoumbos, 2008). In multi-chart settings, the IC RL distribution depends on a vector of control limits. Finding appropriate values of the control limit vector thus requires solving a more complicated design, which often includes a constraint on the run lengths characteristics of the individual control charts.

In cases when analytical and deterministic numerical methods are unavailable to determine control limits of a control chart, methods based on Monte Carlo simulations offer a popular alternative. The only requirement for Monte-Carlo-based methods is to be able to simulate IC run lengths. This is typically achieved by generating new data from either the IC process distribution, if it is assumed known, or by approximations such as the bootstrap (Gandy and Kvaløy, 2013).

Among Monte-Carlo-based methods, a popular technique to determine the control limits is the bisection searching algorithm (cf., Qiu, 2008; Dickinson et al., 2014; Lai et al., 2023). This approach leverages the fact that a typical IC RL characteristic of interest (e.g., ARL_0) is a monotone function of a control limit. Thus, the bisection searching algorithm would converge, provided that the pre-specified initial interval for searching is wide enough to contain the solution. Typically, a wide initial interval is chosen, which leads to the computation of some large run length values during the searching process (Capizzi and Masarotto, 2016). This can result in a significant computational cost to find the solution. Some authors use a preliminary search phase to find the suitable interval for the bisection search (Bizuneh and Wang, 2019). Nevertheless, these approaches are designed for the specific control chart in use and may not be readily applicable to different control charts. In addition, application of the bisection algorithm in settings with multiple control charts can be computationally challenging.

A different approach to find the control limits is the more recent Stochastic Approximation (SA) algorithm discussed originally in Capizzi and Masarotto (2016), which uses SA methods (Robbins and Monro, 1951; Kushner and Yin, 2003) to implement a stochastic gradient descent iterative algorithm that converges to the desired control limit values. While efficient and developed for multi-chart settings, the SA algorithm requires pre-specification of a large number of tuning parameters. Although the recommended parameter settings in Capizzi and Masarotto (2016) are typically robust for single-chart designs, in our experience, parameter tuning is required in the multi-chart scenario so as not to let the early iterations diverge from the solution. In cases when early gradient descent iterations move the candidate control limit value far away from the solution, the SA algorithm requires a large number of iterations to reach convergence.

To address the computational challenges discussed above, this paper presents a modified bisection algorithm, which is henceforth referred to as the BA–Bisection algorithm, where "BA" stands for "bootstrap-assisted" for the reason given below. The major goal of the BA–Bisection algorithm is to overcome the shortcomings of the traditional bisection searching algorithm discussed above and extend its applicability to multi-chart scenarios. Rather than approximating the IC RL distribution, as in the traditional bisection searching algorithm, its key idea is to simulate the IC distribution of the charting statistic at each observation time during process monitoring. Then, the traditional bisection searching algorithm can be applied to the estimated IC distribution of the charting statistic with minimal computational cost to find the appropriate control limit values. A similar idea was adopted in Chatterjee and Qiu (2009), where bootstrap was used to approximate the IC distribution of the CUSUM charting statistic conditionally on the elapsed time T_n (also called *spring length*) since the statistic was last set to zero. See also Qiu and Xie (2022). This was then used to define a sequence of control limits for the CUSUM charting statistic conditionally on T_n. However, their method was designed specifically for CUSUM charts, and extensions to other types of control charts or multi-chart settings are not easy.

The BA–Bisection method does not require an initial search interval to be specified, making it suitable for software used to automatically determine control limits for various control charts and data distributions. Additionally, the proposed method can be extended to the multi-chart scenario due to its computational efficiency.

The remaining parts of the paper are organized as follows. Section 2 discusses the BA–Bisection algorithm in detail and illustrates its advantages over the traditional bisection approach. By leveraging its computational advantages, an extension of the BA–Bisection algorithm is also proposed to address the multi-chart scenario. Section 3 presents some simulation results to assess the performance of the proposed methodology. First, in Section 3.1, the BA–Bisection algorithm is compared to the traditional bisection searching algorithm. Then, in Section 3.2, the proposed method is compared to the SA algorithm for designing multi-chart schemes. In Section 4, an example using a recently-developed control chart for monitoring high-dimensional partially-observed data streams (Xian et al., 2018) illustrates the applicability of the proposed method to a complex scenario. Finally, Section 5 offers some concluding remarks.

2 Methodology

Since determination of the control limits of a control chart is under the IC condition, all discussions in this section are under that condition if there is no further specification. Let C_t be the value of the charting statistic at time $t \geq 1$, and $\boldsymbol{C} = (C_1, C_2, \ldots)$ be the *trajectory* of the charting statistic. For simplicity, let us assume that $C_t \geq 0$ (e.g., the upward CUSUM chart or the multivariate EWMA chart) and a signal would be given by the chart if its charting statistic exceeds a control limit $h > 0$. Then, $\mathrm{RL}(\boldsymbol{C}, h) = \inf\{t > 0 : C_t > h\}$ is the run length of the control chart, which depends on the IC distribution of the charting statistic and h. Let $G_0(\mathrm{RL}(\boldsymbol{C}, h))$ be a specific characteristic of interest of the RL distribution, such as ARL_0, MRL_0, or some quantiles of the IC RL distribution. In the current setup, it is obvious that G_0 is a non-decreasing function of h.

The design of a typical control chart involves finding the value h^* of the control limit h such that $G_0(\mathrm{RL}(\boldsymbol{C}, h)) = a$, where $a > 1$ is a pre-determined nominal value of $G_0(\mathrm{RL}(\boldsymbol{C}, h))$. In order to achieve this property using Monte-Carlo simulations, the classical approach is to estimate $G_0(\mathrm{RL}(\boldsymbol{C}, h))$ with a large number M of simulation runs. Then, a bisection searching algorithm is used to search for the value of h^*. More specifically, from each simulation run, a RL value can be recorded. Thus, $G_0(\mathrm{RL}(\boldsymbol{C}, h))$ can be estimated using the M simulated RL values (Qiu, 2013). Then, the bisection searching algorithm searches for h^* in a pre-specified initial interval $[h_L, h_U]$ satisfying the conditions that $G_0(\mathrm{RL}(\boldsymbol{C}, h_L)) \leq a \leq G_0(\mathrm{RL}(\boldsymbol{C}, h_U))$.

This traditional approach has some drawbacks, as discussed in Section 1. Most notably, determination of the initial interval $[h_L, h_U]$ depends on the specific control chart, and can be difficult to specify in some complicated settings. Therefore, a very wide initial interval is typically used in applications, which would result in a large number of "useless" run lengths (i.e., $\mathrm{RL}(\boldsymbol{C}, h) \gg a$) being simulated during the search process (Capizzi and Masarotto, 2016). Consequently, a significant amount of computing is wasted. In addition, extending the traditional bisection searching algorithm to multi-chart settings is typically complex and computationally intensive. To address these issues, we propose a modification of the traditional bisection searching algorithm, which is described in detail below.

2.1 Proposed BA–Bisection algorithm

The key idea in our proposed modification is to approximate the IC distribution of the charting statistic C_t at each $t \geq 1$, rather than the IC distribution of the RL. Denote by $\boldsymbol{C}_i^* = \{C_{i,t}^*\}_{t=1}^T$ the i-th simulated trajectory of the charting statistic, where T is a pre-specified maximum time after which the process monitoring is ignored. The value of T is usually chosen to be large, so as not to introduce excessive bias in the resulting control limit estimate. If $G_0(\mathrm{RL}(\boldsymbol{C}, h))$ is ARL_0, typical choices are $T \geq 10 \cdot a$ (Qiu and Xie, 2022; Xie and Qiu, 2023a). In this paper, $T = 10 \cdot a$ has been used in all numerical studies. Then, prior to the application of the bisection searching algorithm, a total of M trajectories $\mathcal{C} = \{\boldsymbol{C}_1^*, \boldsymbol{C}_2^*, \ldots, \boldsymbol{C}_M^*\}$ has been simulated from the IC process distribution. This can be done by either sampling from the true IC process distribution if it is known, or from a reference IC sample by a bootstrap or other resampling procedures.

The proposed BA–Bisection algorithm can be described as follows. In the kth iteration, for $k = 1, 2, \ldots, K$, the IC characteristic $G_0(\mathrm{RL}(\boldsymbol{C}, h^{(k)}))$ using the control limit value $h^{(k)}$ is approximated by

$$\widehat{G_0}(\mathrm{RL}(\boldsymbol{C}, h^{(k)})) = \int G_0(\mathrm{RL}(\boldsymbol{C}, h^{(k)}))\ d\widehat{P}^*(\boldsymbol{C}), \tag{1}$$

where $\widehat{P}^*$ is the empirical distribution of the simulated trajectories $\mathcal{C}$. As an example, if G_0 is just ARL_0, then, the estimate of the ARL_0 using the control limit $h^{(k)}$ can be approximated by

$$\widehat{\mathrm{ARL}_0}(h^{(k)}) = \frac{1}{M}\sum_{i=1}^{M} \mathrm{RL}(\boldsymbol{C}_i^*, h^{(k)}). \tag{2}$$

Naturally, substituting the empirical average in (2) with the empirical median or the qth empirical quantile yields the estimates of MRL_0 and the qth quantile of the IC RL distribution, respectively. The iterative algorithm is terminated whenever

$$\left|\widehat{G_0}(\mathrm{RL}(\boldsymbol{C}, h^{(k)})) - a\right| < \varepsilon_1 \quad \text{or} \quad \left|h^{(k+1)} - h^{(k)}\right| < \varepsilon_2,$$

where $\varepsilon_1, \varepsilon_2 > 0$ are two pre-specified small values. Due to the analogies between the proposed method and the classical bootstrap procedure, our proposed algorithm is referred to as the Bootstrap–Assisted Bisection (BA–Bisection) algorithm.

As a comparison, in the traditional bisection searching algorithm, the quantity in (1) is approximated by simulating M new run lengths in *each* iteration $k = 1, 2, \ldots, K$. The key advantage of our modified algorithm lies in its computational efficiency. Specifically, the bulk of the computational resources is used to generate the set of trajectories $\mathcal{C}$. Once these have been generated, applying the bisection search to determine the required control limit becomes computationally trivial. This allows for a substantial improvement in computational cost, especially when the charting statistic is computationally expensive to evaluate.

Additionally, our proposed modification eliminates the requirement to pre-specify an initial interval $[h_L, h_U]$ within which the control limit is searched. Instead, this interval can be determined based on the simulated trajectories in $\mathcal{C}$. Specifically, if $T > a$ (which is always true in reality), then the initial interval can be defined to be

$$[h_L, h_U] = \left[\min_{C_{i,t}^* \in \mathcal{C}} C_{i,t}^*, \max_{C_{i,t}^* \in \mathcal{C}} C_{i,t}^*\right],$$

and the property $G_0(\mathrm{RL}(\boldsymbol{C}, h_L)) \leq a \leq G_0(\mathrm{RL}(\boldsymbol{C}, h_U))$ can be verified easily.

It is worth mentioning that control charts can sometimes use dynamic (or time-varying) control limits of the form

$$h(t) = h \cdot g(t), \tag{3}$$

where $g(t)$ is a known function of t. An important example is the EWMA chart for detecting process mean shifts, where the dynamic control limit is defined to be

$$h(t) = \rho \cdot \sqrt{\frac{\lambda}{2-\lambda}\left[1-(1-\lambda)^{2t}\right]}.$$

and ρ is a constant. The BA–Bisection algorithm can be modified as follows to handle such cases by searching for the appropriate value of ρ in the initial interval

$$[\rho_L, \rho_U] = \left[\frac{1}{A} \min_{C^*_{i,t} \in \mathcal{C}} C^*_{i,t}, \frac{1}{B} \max_{C^*_{i,t} \in \mathcal{C}} C^*_{i,t}\right],$$

where

$$A = \max_{1 \le t \le T} g(t), \text{ and } B = \min_{1 \le t \le T} g(t).$$

Algorithm 1 below provides a pseudo-code for the proposed BA–Bisection approach.

2.2 Extension to multi-chart cases

Multi-chart monitoring schemes are characterized by the simultaneous application of $J > 1$ control charts. The jth charting statistic is compared to the control limit h_j, for $j = 1, \ldots, J$. For simplicity, it is assumed that the joint process monitoring scheme triggers an alarm whenever one of the J charting statistics is larger than its control limit. Let RL_j indicate the run length of the jth control chart and $\mathrm{RL} = \min\{\mathrm{RL}_1, \mathrm{RL}_2, \ldots, \mathrm{RL}_J\}$ be the run length of the joint monitoring scheme. Traditionally, the vector of J control limits, $\boldsymbol{h} = (h_1, h_2, \ldots, h_J)$, of a multi-chart

Algorithm 1 BA–Bisection algorithm

Input: M, K, T, A, B, a, $\varepsilon_1 > 0$, $\varepsilon_2 > 0$

```
1: Simulate 𝒞 = {C*_1, C*_2, ..., C*_M}
2: h_L ← min_{C*_{i,t} ∈ 𝒞} C*_{i,t}/A, h_U ← max_{C*_{i,t} ∈ 𝒞} C*_{i,t}/B.
3: for k = 1, ..., K do
4:     h^(k) ← (h_U + h_L)/2
5:     Calculate Ĝ_0(RL(C, h^(k))) using Equation (1).
6:     if Ĝ_0(RL(C, h^(k))) > a then
7:         h_U ← h^(k)
8:     else
9:         h_L ← h^(k)
10:    end if
11:    if |Ĝ_0(RL(C, h^(k))) − a| < ε_1 or |h^(k) − h^(k−1)| < ε_2 then
12:        break
13:    end if
14: end for
15: return h^(k)
```

monitoring scheme is selected to satisfy the following conditions:

$$\begin{cases} G_0(\mathrm{RL}(\boldsymbol{C}, \boldsymbol{h})) = a, \\ G_0(\mathrm{RL}_1(\boldsymbol{C}, h_1)) = G_0(\mathrm{RL}_2(\boldsymbol{C}, h_2)) = \ldots = G_0(\mathrm{RL}_J(\boldsymbol{C}, h_J)), \end{cases} \tag{4}$$

where $a > 0$ is a pre-specified nominal value for $G_0(\mathrm{RL}(\boldsymbol{C}, \boldsymbol{h}))$. The above constraint on the run lengths of individual control charts is enforced so that no specific control chart is favored over another. However, weighting mechanisms can be easily incorporated in the constraint if individual control charts differ in their importance.

To accommodate the constraints in (4) within the framework of the BA–Bisection algorithm, an adaptation of Algorithm 1 is formulated below. This adaptation is inspired by the nested secant algorithm of Knoth and Morais (2015), which was used in the single-chart scenario to determine upper and lower control limits when monitoring asymmetric process distributions. The key property exploited here is the low computational cost of applying the bisection search once the M replications of the IC trajectories $\boldsymbol{C}_i^* = \{C_{i,t,j}^* : j = 1, \ldots, J\}_{t=1}^T$, for $i = 1, \ldots, M$, have been generated, where $C^*_{i,t,j}$ denotes the ith simulated value of the jth charting statistic at time t. In the general case of a time-varying control limit (3), let $\mathcal{C}_j = \{C^*_{i,t,j}, i = 1, \ldots, M,\ t = 1, \ldots, T\}$ be the simulated trajectories of the jth charting statistic, and

$$\left[h^{(0)}_{L,j}, h^{(0)}_{U,j}\right] = \left[\min_{C^*_{i,t} \in \mathcal{C}_j} C^*_{i,t}/A_j, \max_{C^*_{i,t} \in \mathcal{C}_j} C^*_{i,t}/B_j\right],$$

be the corresponding initial interval for search, for $j = 1, \ldots, J$. In the above equation,

$$A_j = \max_{1 \le t \le T} g_j(t), \text{ and } B_j = \min_{1 \le t \le T} g_j(t),$$

where $g_j(t)$ is the function that defines the time-varying control limit for the j-th control chart. Then, the modified algorithm for designing multi-chart monitoring schemes can be described below:

1. In the kth iteration, for $k = 1, 2, \ldots, K$, the control limits $\{h_1, h_2, \ldots, h_J\}$ are updated by the bisection search criterion using the estimated joint IC ARL value at the $(k-1)$th iteration.
2. Then, the IC RL characteristic $G_0(\mathrm{RL}_1(\boldsymbol{C}, h^{(k)}))$ of the first chart is estimated using Equation (1).
3. The algorithm then determines the control limits h_j such that $G_0(\mathrm{RL}_j(\boldsymbol{C}, h^{(k)})) = G_0(\mathrm{RL}_1(\boldsymbol{C}, h^{(k)}))$, for all $j = 2, \ldots, J$. This step is carried out using the BA–Bisection algorithm described in Section 2.1 with initial search intervals $[h^{(k-1)}_{L,j}, h^{(k-1)}_{U,j}]$.
4. Finally, the exit criteria are checked. Return to Step 1 if none of them are met. Exit and report the searched control limit values otherwise.

Algorithm 2 provides the pseudo-code of the extended BA–Bisection algorithm for handling multi-chart designs.

Algorithm 2 BA–Bisection algorithm for multi-chart designs

Input: M, M, K, T, a, $\varepsilon_1 > 0$, $\varepsilon_2 > 0$

Simulate $\mathcal{C} = \{\boldsymbol{C}_1^*, \boldsymbol{C}_2^*, \ldots, \boldsymbol{C}_M^*\}$

$h_{L,j} \leftarrow \min_{C_{i,t}^* \in \mathcal{C}_j} C_{i,t}^*/A_j$, $h_{U,j} \leftarrow \max_{C_{i,t}^* \in \mathcal{C}_j} C_{i,t,j}^*/B_j$ for $j = 1, \ldots, J$.

for $k = 1, \ldots, K$ **do**

  $h_1^{(k)} \leftarrow (h_{U,1} + h_{L,1})/2$

  Estimate $\widehat{G_0}(\mathrm{RL}_1(\boldsymbol{C}, h_1^{(k)}))$ using Equation (1).

  for $j = 2, \ldots, J$ **do**

    Find $h_j^{(k)}$ such that $G_0(\mathrm{RL}_j(\boldsymbol{C}, h_j^{(k)})) = \widehat{G_0}(\mathrm{RL}_1(\boldsymbol{C}, h_1^{(k)}))$ using Algorithm 1 and the set of simulated trajectories $\mathcal{C}$.

  end for

  Calculate $G_0(\mathrm{RL}(\boldsymbol{C}, \boldsymbol{h}^{(k)}))$ using Equation (1).

  if $\widehat{G_0}(\mathrm{RL}(\boldsymbol{C}, \boldsymbol{h}^{(k)})) > a$ **then**

    $\boldsymbol{h}_U \leftarrow \boldsymbol{h}^{(k)}$

  else

    $\boldsymbol{h}_L \leftarrow \boldsymbol{h}^{(k)}$

  end if

  if $|\widehat{G_0}(\mathrm{RL}(\boldsymbol{C}, \boldsymbol{h}^{(k)})) - a| < \varepsilon_1$ **or** $\|\boldsymbol{h}^{(k)} - \boldsymbol{h}^{(k-1)}\| < \varepsilon_2$ **then**

    break

  end if

end for

return $\boldsymbol{h}^{(k)}$

The following proposition ensures that Algorithm 2 can indeed find reasonable values for the control limits. Its proof is given in Section A.

Proposition 1. *If $G_0(RL(\boldsymbol{C}, \boldsymbol{h}))$ is a non-decreasing function of each element of $\boldsymbol{h}$ and G_0 can be calculated accurately, then the solution from Algorithm 2 satisfies the constraints in Equation (4).*

3 Simulation Results

In this section, we evaluate the numerical performance of the proposed BA–Bisection algorithm using various control charts. First, the proposed BA–Bisection algorithm is compared with the traditional bisection searching algorithm in terms of accuracy and computing time when determining the control limit values. Then, the proposed BA–Bisection algorithm is compared with the SA algorithm using various multi-chart monitoring schemes. The results presented here were obtained on a Red Hat Enterprise Linux release 8.7 machine with 2GHz Intel Xeon Gold 6348H CPUs.

3.1 Comparison with the traditional bisection searching algorithm

In this part, the proposed BA–Bisection algorithm is compared with the traditional bisection searching algorithm using the following control charts for monitoring the mean of a p-dimensional process that has i.i.d. IC Gaussian observations:

1. A MEWMA chart (Crowder, 1989) when $p = 3$, and the smoothing matrix is set to be $\Lambda = \mathrm{diag}(0.2, 0.2, 0.2)$.
2. A MCUSUM chart (Crosier, 1988) when $p = 5$, and the allowance constant is set to be $k = 0.25$.
3. A distribution-free CUSUM chart (Qiu, 2008) based on data categorization using the IC medians when $p = 3$, and the allowance constant is set to be $k = 1$. Its RL values are simulated as suggested in Qiu (2008) by generating data from the categorized process using its IC probability distribution.

We consider the design of the three control charts by setting the nominal value of the IC RL characteristic to be $a = 200$ in the following two scenarios: i) $G_0(\mathrm{RL}(\boldsymbol{C}, h))$ is ARL_0, and ii) $G_0(\mathrm{RL}(\boldsymbol{C}, h))$ is MRL_0. For both algorithms, different values of M are considered to examine their impact on the computational cost. For both algorithms, the tolerance parameters are set to be $\varepsilon_1 = 1$, $\varepsilon_2 = 10^{-6}$, and the RL values are capped at $T = 2000$. The traditional bisection searching algorithm is run using the initial interval $[h_L, h_U] = [0, 100]$ for all control charts. Table 1 and Table 2 present the results for $G_0 = \mathrm{ARL}_0$ and $G_0 = \mathrm{MRL}_0$, respectively, based on 100 replicated simulations of the control limit search. In each search, after the control limit value h is determined, the value of G_0 is estimated using 10^5 simulated RL values.

Table 1: Searched control limit (h) values of several control charts computed via the proposed BA–Bisection algorithm and the traditional bisection searching algorithm in cases when G_0 = ARL$_0$. The table displays means with standard deviations in parentheses of the related quantities based on 100 replicated simulations.

	M	time (s)	ARL$_0$	h
		MEWMA		
BA-Bisection				
	1000	0.755 (0.081)	200.090 (6.498)	11.866 (0.077)
	5000	3.823 (0.391)	199.653 (2.823)	11.863 (0.033)
	10000	7.737 (0.811)	199.811 (1.918)	11.864 (0.023)
	25000	19.383 (2.044)	199.909 (1.492)	11.864 (0.017)
Bisection				
	1000	2.742 (0.603)	200.236 (4.244)	11.868 (0.051)
	5000	11.865 (2.474)	200.332 (2.475)	11.870 (0.029)
	10000	22.918 (4.124)	199.861 (1.947)	11.865 (0.022)
	25000	55.375 (7.078)	199.965 (1.207)	11.865 (0.012)
		MCUSUM		
BA-Bisection				
	1000	1.660 (0.151)	199.274 (5.539)	14.789 (0.099)
	5000	8.265 (0.743)	199.882 (2.271)	14.801 (0.041)
	10000	16.554 (1.447)	199.989 (2.015)	14.804 (0.036)
	25000	41.516 (3.574)	200.184 (1.204)	14.808 (0.020)
Bisection				
	1000	5.654 (1.267)	199.988 (3.722)	14.805 (0.067)
	5000	25.031 (4.437)	200.491 (2.105)	14.813 (0.036)
	10000	47.766 (6.994)	200.016 (1.625)	14.807 (0.027)
	25000	118.352 (13.283)	200.093 (1.280)	14.807 (0.019)
		Distribution-free CUSUM		
BA-Bisection				
	1000	86.578 (3.340)	199.841 (6.572)	12.172 (0.085)
	5000	430.578 (16.411)	199.985 (3.040)	12.176 (0.076)
	10000	859.844 (32.569)	200.622 (2.160)	12.182 (0.072)
	25000	2139.365 (83.603)	200.470 (1.610)	12.181 (0.070)
Bisection				
	1000	283.607 (57.736)	200.514 (4.447)	12.181 (0.074)
	5000	1244.497 (200.503)	200.268 (2.633)	12.179 (0.073)
	10000	2404.964 (245.073)	199.659 (1.834)	12.174 (0.069)
	25000	5976.661 (419.766)	200.269 (1.386)	12.180 (0.068)

From the tables, it can be seen that the proposed BA–Bisection algorithm is much faster than the traditional bisection searching algorithm while providing a similar degree of accuracy for the searched control limit values in all cases considered. As seen in Tables 1 and 2, computation of the control limits using the traditional bisection searching algorithm could be expensive, and the computational cost can be reduced

Table 2: Searched control limit (h) values of several control charts computed via the proposed BA–Bisection algorithm and the traditional bisection searching algorithm in cases when $G_0 = \text{MRL}_0$. The table displays means with standard deviations in parentheses of the related quantities based on 100 replicated simulations.

method	M	time (s)	MRL_0	h
		MEWMA		
BA-Bisection				
	1000	0.752 (0.083)	201.130 (9.294)	12.736 (0.109)
	5000	3.836 (0.385)	199.320 (4.175)	12.716 (0.049)
	10000	7.751 (0.750)	199.800 (3.204)	12.720 (0.036)
	25000	19.394 (1.913)	200.140 (1.912)	12.726 (0.020)
Bisection				
	1000	4.267 (0.979)	201.110 (6.549)	12.735 (0.075)
	5000	20.322 (4.291)	199.830 (2.613)	12.721 (0.030)
	10000	39.990 (8.282)	200.090 (2.283)	12.724 (0.023)
	25000	94.317 (15.510)	199.970 (1.594)	12.721 (0.016)
		MCUSUM		
BA-Bisection				
	1000	1.650 (0.149)	201.550 (6.601)	15.940 (0.119)
	5000	8.285 (0.645)	199.565 (3.472)	15.904 (0.062)
	10000	16.618 (1.277)	199.620 (2.469)	15.906 (0.046)
	25000	42.465 (3.994)	200.270 (1.734)	15.915 (0.031)
Bisection				
	1000	7.387 (1.815)	200.795 (5.590)	15.927 (0.104)
	5000	32.424 (7.753)	200.250 (2.472)	15.914 (0.045)
	10000	61.631 (14.222)	199.995 (1.782)	15.913 (0.029)
	25000	151.042 (27.429)	199.970 (1.167)	15.911 (0.018)
		Distribution-free CUSUM		
BA-Bisection				
	1000	95.149 (8.052)	201.540 (9.951)	13.051 (0.187)
	5000	477.569 (41.588)	199.915 (4.388)	13.034 (0.152)
	10000	956.087 (81.867)	199.990 (3.112)	13.034 (0.145)
	25000	2383.169 (205.662)	200.400 (2.507)	13.036 (0.143)
Bisection				
	1000	470.683 (83.921)	199.420 (5.961)	13.027 (0.165)
	5000	2484.026 (472.182)	199.870 (3.212)	13.035 (0.147)
	10000	4822.338 (946.967)	199.690 (2.497)	13.033 (0.140)
	25000	11187.710 (1918.285)	200.380 (1.698)	13.038 (0.137)

significantly by using the proposed BA–Bisection algorithm while providing a similar degree of accuracy.

3.2 Comparison with the SA algorithm for designing multi-chart monitoring schemes

In this part, we compare the proposed BA–Bisection algorithm with the SA algorithm Capizzi and Masarotto (2016) for designing a multi-chart monitoring scheme that meets the constraints in Equation (4). The comparison considers the same IC process distribution considered in Section 3.1, and the following multi-chart monitoring schemes are considered:

1. A multi-chart scheme based on the combination of four univariate EWMA control charts (Roberts, 1959) when $p = 1$ with the weighting parameters set to be 0.05, 0.1, 0.2, and 0.5, respectively.
2. A multi-chart scheme combining the Hotelling's T^2 chart (Mason and Young, 2002; Montgomery, 2020) and three MCUSUM control charts when $p = 5$ with allowance constants set to be 0.1, 0.25, and 0.5, respectively.
3. A multi-chart scheme by using two univariate distribution-free CUSUM charts, when $p = 2$ with the allowance constants set to be 0.1 and 0.5, respectively.

In this simulation, we set the nominal value $a = 200$ and consider the search of the control limits for all the multi-chart schemes when $G_0(\mathrm{RL}(\boldsymbol{C}, \boldsymbol{h})) = \mathrm{ARL}_0$ or $G_0(\mathrm{RL}(\boldsymbol{C}, \boldsymbol{h})) = \mathrm{MRL}_0$. We set $M = 10000$ for the proposed BA–Bisection algorithm, and the tolerance parameters are set to be $\varepsilon_1 = 1$ and $\varepsilon_2 = 10^{-3}$. The SA algorithm is used with its parameter values recommended by Capizzi and Masarotto (2016). The accuracy parameter $\gamma = 0.01$ of the SA algorithm is used, which is an intermediate value between the low-accuracy and high-accuracy choices considered in their simulation study. Additionally, we fine-tune the parameter $A_{\max}$ of the SA algorithm to ensure convergence within reasonable time frames for each control chart. Thus, the simulation study presented here favours the SA algorithm over the proposed BA–Bisection algorithm, since the latter does not require selection of such tuning

Table 3: Searched control limit values of several multi-chart schemes computed via the proposed BA–Bisection algorithm and the SA algorithm when $G_0 = \text{ARL}_0$ or MRL_0. The table displays means with standard errors in parentheses of the indicated quantities based on 100 replicated simulations. For each case, the best average computing times are indicated in bold.

	$G_0 = \text{ARL}_0$		$G_0 = \text{MRL}_0$	
	BA–Bisection	SA	BA–Bisection	SA
		Multiple EWMA		
time (s)	**260.937** (115.763)	1223.064 (242.84)	**370.747** (111.832)	2520.891 (1148.35)
G_0(RL)	200.106 (2.020)	203.864 (1.678)	200.355 (3.439)	204.480 (2.935)
G_0(RL$_1$)	407.691 (5.546)	420.696 (7.077)	419.685 (7.684)	430.200 (5.350)
G_0(RL$_2$)	407.751 (5.147)	411.695 (1.804)	419.90 (8.084)	429.050 (5.912)
G_0(RL$_3$)	407.983 (5.499)	417.093 (4.586)	420.52 (9.302)	428.465 (6.65)
G_0(RL$_4$)	407.662 (5.274)	414.143 (2.905)	419.815 (9.118)	428.785 (7.39)
h_1	0.405 (0.001)	0.407 (0.001)	0.430 (0.001)	0.432 (0.001)
h_2	0.628 (0.001)	0.629 (0.000)	0.661 (0.002)	0.663 (0.001)
h_3	0.964 (0.002)	0.967 (0.001)	1.008 (0.002)	1.011 (0.002)
h_4	1.737 (0.002)	1.739 (0.001)	1.806 (0.003)	1.810 (0.003)
		T^2 and multiple MCUSUM		
time (s)	**280.015** (66.046)	2228.464 (751.684)	**302.637** (73.588)	3623.74 (681.274)
G_0(RL)	199.972 (1.986)	200.598 (0.562)	200.260 (2.485)	200.550 (0.947)
G_0(RL$_1$)	493.311 (6.935)	493.072 (1.860)	468.235 (8.328)	457.895 (2.254)
G_0(RL$_2$)	487.451 (5.414)	493.606 (1.734)	468.930 (7.327)	504.360 (1.755)
G_0(RL$_3$)	490.067 (6.695)	493.418 (1.913)	468.410 (7.976)	474.325 (2.192)
G_0(RL$_4$)	492.432 (6.351)	493.592 (1.992)	469.115 (8.261)	463.340 (2.438)
h_1	18.877 (0.031)	18.875 (0.006)	19.607 (0.039)	19.553 (0.004)
h_2	29.622 (0.098)	29.736 (0.019)	31.653 (0.141)	32.341 (0.014)
h_3	18.024 (0.047)	18.048 (0.007)	19.017 (0.059)	19.062 (0.006)
h_4	10.879 (0.020)	10.881 (0.004)	11.362 (0.027)	11.343 (0.003)
		Multiple distribution-free CUSUM		
time (s)	**393.793** (68.299)	3886.558 (2597.667)	**387.281** (69.074)	7516.773 (5718.945)
G_0(RL)	199.915 (2.318)	200.412 (0.820)	200.010 (3.704)	200.960 (1.253)
G_0(RL$_1$)	325.853 (4.063)	327.259 (1.314)	328.575 (6.735)	326.845 (2.037)
G_0(RL$_2$)	326.801 (4.118)	326.793 (1.319)	329.765 (6.347)	334.560 (2.529)
h_1	8.893 (0.041)	8.899 (0.034)	9.646 (0.063)	9.635 (0.048)
h_2	9.127 (0.035)	9.128 (0.023)	9.851 (0.028)	9.874 (0.016)

parameters. When MRL_0 is the IC RL characteristic of interest, the SA algorithm is applied using the gradient descent iteration described in Capizzi and Masarotto (2009). All results reported here are obtained from 100 replicated simulations of the control limit search. In each simulation, the ARL_0 or MRL_0 values are estimated using 10^5 simulated RL values once the control limits are determined.

Table 3 shows the results for both algorithms. From the table, it can be seen that the BA–Bisection algorithm can successfully find the control limits that satisfy

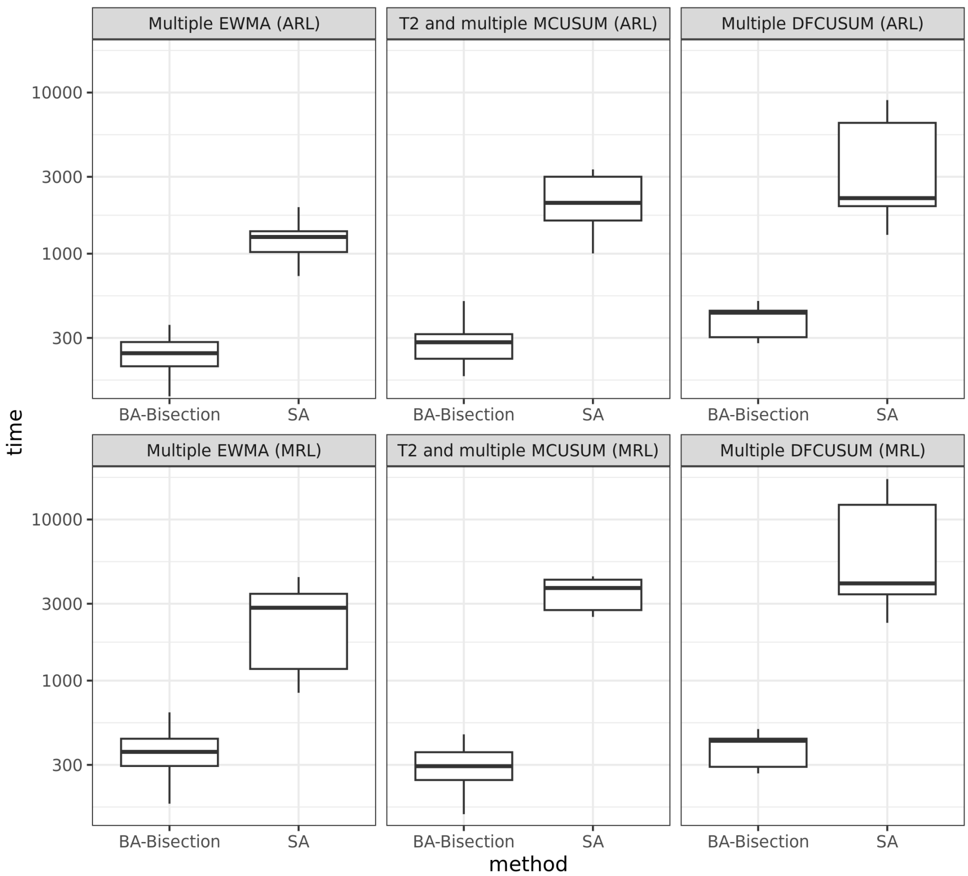

Fig. 1: Computing times of the BA–Bisection and the SA algorithm for the three considered multi-charts when $G_0 = \text{ARL}_0$ (top row) and $G_0 = \text{MRL}_0$ (bottom row). Results are based on 100 independent simulations, and are displayed on a log scale.

the constraints in Equation (4) in all cases considered. Furthermore, its solution is obtained at a small fraction of the computational cost of the SA algorithm. In cases when MRL_0 is concerned, the difference in computing times of the two algorithms is remarkably large.

Figure 1 illustrates the computing times of the two algorithms. From the figure, it can be seen that the computing cost of the BA–Bisection algorithm is substantially lower, especially when considering the multiple distribution-free CUSUM chart. This may be a result of the discreteness of the monitoring statistics, which makes the

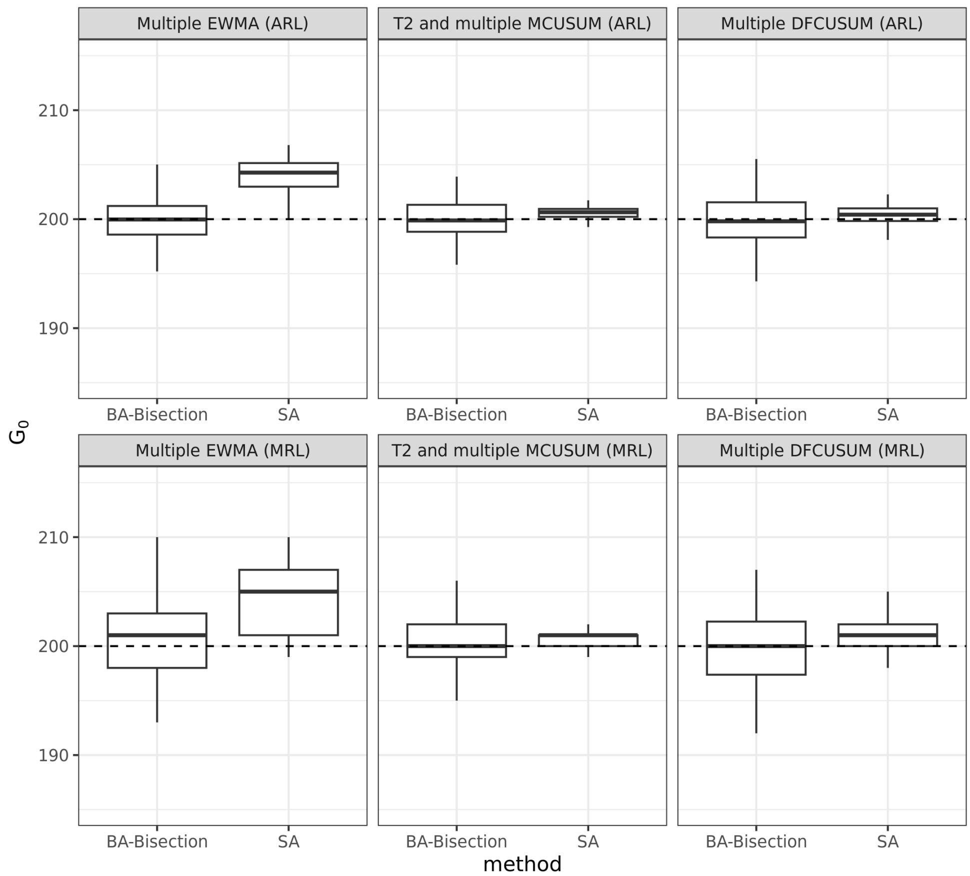

Fig. 2: Estimated values of G_0 using the solutions obtained by the BA–Bisection and the SA algorithm when $G_0 = \text{ARL}_0$ (top row) and $G_0 = \text{MRL}_0$ (bottom row). Results are based on 100 replicated simulations, and each estimate is based on 10^5 simulated run lengths. The dashed black line indicates the nominal value of G_0.

application of gradient-based approaches more difficult. Additionally, Figure 2 displays the estimated value of G_0 using the solution obtained by the two algorithms. The BA–Bisection algorithm appears to be more accurate when $G_0 = \text{MRL}_0$ compared to the SA algorithm. In this case, the solutions obtained by the SA algorithm have slightly larger values of MRL_0 than the nominal value of 200. The same behavior can also be seen in the multiple EWMA case when $G_0 = \text{ARL}_0$. These results suggest that the proposed extension of the proposed BA–Bisection algorithm presents a competitive alternative to the SA algorithm for designing multi-chart monitoring schemes.

4 Improving the Efficiency Using Parallel Computation

Whenever a high level of precision in estimating the control limits is required, the computational demand could be substantial. In contemporary computing environments, availability of multiple central processing units (CPUs) presents an avenue for cost-efficient computation through parallelization. To this end, the proposed BA–Bisection algorithm is inherently parallelizable. Specifically, the initial generation of the set of trajectories $\mathcal{C}$ can be efficiently distributed across multiple CPUs. This parallelization strategy is characterized by its simplicity of implementation and no coordination requirements among the CPUs engaged in the computation process.

Here, we present an example to illustrate the application of the proposed BA–Bisection algorithm in a high-dimensional setting using parallel computation. Let us consider the R-SADA control chart introduced recently by Xian et al. (2018) for monitoring partially-observed data streams. This control chart depends on two parameters: 1) the CUSUM chart allowance constant k, and 2) the minimum shift $\mu_{\min}$ to be detected by the control chart. The choice of appropriate values of k and $\mu_{\min}$ depends on the underlying process distribution and the number of observable data streams. Therefore, even with prior information on the shift to be detected, an appropriate choice of parameters might be challenging. On the other hand, a multi-chart scheme using different combinations of the parameters could have satisfactory performance in various OC scenarios. As a demonstration, let us consider a multi-chart monitoring scheme consists of four R-SADA control charts with the following four sets of parameters, respectively:

$$1)\quad k = 0.5, \mu_{\min} = 1.5; \quad 2)\quad k = 0.75, \mu_{\min} = 2.0;$$

$$3)\quad k = 1.0, \mu_{\min} = 2.5; \quad 4)\quad k = 1.25, \mu_{\min} = 3.0.$$

The multi-chart monitoring scheme is then used to monitor $p = 200$ independent Normally-distributed data streams, of which $q = 20$ are observable at each observation time. The results presented in Table 4 are based on 100 replicated simulations. For each replication, the trajectories $\mathcal{C}$ are computed using one CPU, and in parallel using four and eight CPUs. For each control limit value, the values of G_0 are calculated using 10^4 simulated RL values. From the table, it can be seen that the proposed BA–Bisection algorithm can find a satisfactory solution in both cases when ARL_0 and MRL_0 are considered. Furthermore, computing times are kept in a reasonable range by allowing parallel computation in generating the IC trajectories $\mathcal{C}$.

5 Conclusions

In this paper, we have introduced a modified version of the traditional bisection searching algorithm, termed BA–Bisection algorithm, designed for determining control limits of control charts. It has been has shown that the proposed method significantly improves the computational efficiency of the traditional bisection searching algorithm while maintaining a comparable level of accuracy. Leveraging this enhanced computational efficiency, an extension of the algorithm has also been proposed to design multi-chart monitoring schemes.

The proposed approach has been compared to the traditional bisection searching algorithm and a recent SA algorithm designed for multi-chart applications (Capizzi and Masarotto, 2016). The results show that the proposed method is computationally efficient, compared to the traditional bisection searching algorithm while maintaining

Table 4: Searched control limit values of the multi-chart scheme combining four R-SADA control charts, computed via the proposed BA–Bisection algorithm when $G_0 = \text{ARL}_0$ or MRL_0. The table displays means with standard deviations in parentheses of the related quantities based on 100 replicated simulations.

	$G_0 = \text{ARL}_0$	$G_0 = \text{MRL}_0$
	Single CPU	
time (s)	2580.348 (67.892)	2578.348 (74.283)
$G_0(\text{RL})$	200.665 (3.192)	200.610 (4.953)
$G_0(\text{RL}_1)$	518.274 (10.358)	543.305 (14.108)
$G_0(\text{RL}_2)$	514.692 (8.674)	543.135 (12.932)
$G_0(\text{RL}_3)$	513.227 (10.178)	543.345 (15.211)
$G_0(\text{RL}_4)$	512.690 (10.418)	543.550 (15.272)
h_1	82.768 (0.234)	88.681 (0.222)
h_2	87.613 (0.149)	91.833 (0.166)
h_3	86.512 (0.180)	90.864 (0.193)
h_4	84.443 (0.203)	89.334 (0.218)
	4 CPUs	
time (s)	1444.891 (50.947)	1442.693 (39.195)
$G_0(\text{RL})$	200.271 (3.328)	200.245 (4.042)
$G_0(\text{RL}_1)$	515.602 (9.023)	544.505 (12.396)
$G_0(\text{RL}_2)$	514.415 (8.594)	543.83 (13.716)
$G_0(\text{RL}_3)$	513.609 (9.043)	544.755 (14.091)
$G_0(\text{RL}_4)$	511.686 (9.414)	543.95 (13.875)
h_1	82.731 (0.204)	88.725 (0.228)
h_2	87.587 (0.156)	91.831 (0.181)
h_3	86.49 (0.176)	90.865 (0.183)
h_4	84.399 (0.19)	89.334 (0.194)
	8 CPUs	
time (s)	1091.088 (57.765)	1054.506 (33.929)
$G_0(\text{RL})$	199.874 (3.361)	200.565 (4.793)
$G_0(\text{RL}_1)$	515.723 (9.711)	545.74 (15.9)
$G_0(\text{RL}_2)$	511.143 (8.222)	540.95 (13.287)
$G_0(\text{RL}_3)$	511.199 (8.869)	544.125 (14.397)
$G_0(\text{RL}_4)$	510.909 (9.192)	543.815 (15.301)
h_1	82.734 (0.223)	88.712 (0.249)
h_2	87.555 (0.151)	91.79 (0.173)
h_3	86.466 (0.164)	90.847 (0.185)
h_4	84.402 (0.192)	89.331 (0.233)

the accuracy of the solution. Compared to the SA algorithm, the results indicate that the BA–Bisection algorithm often achieves similar accuracy with substantially reduced computational costs.

To illustrate the versatility of the method, an example involving online monitoring of high-dimensional partially-observed data streams using a recently-proposed control chart has been presented. This example shows the practical applicability of the proposed method, and harnesses parallel computation to further reduce computational burden.

Our proposed algorithm removes the requirement of the traditional bisection searching algorithm to specify initial intervals for bisection search of the control limits, making the method more convenient to use. In addition, the algorithm appears to be particularly useful when generation of process observations from the IC process distribution is computationally expensive. This may be due to the complexity of the data pre-processing steps, such as data decorrelation, involved in implementation of the related control charts (Qiu and Xie, 2022; Xie and Qiu, 2023b).

Appendix A Proof of Proposition 1

Proof of Proposition 1. Since $G_0(\mathrm{RL}(\boldsymbol{C},\boldsymbol{h}))$ is a non-decreasing function of all components of $\boldsymbol{h}$, in the $(k+1)$th iteration, it holds that

$$\begin{cases} G_0(\mathrm{RL}(\boldsymbol{C},\boldsymbol{h}^{(k+1)})) < G_0(\mathrm{RL}(\boldsymbol{C},\boldsymbol{h}^{(k)})), & \text{if } G_0(\mathrm{RL}(\boldsymbol{C},\boldsymbol{h}^{(k)})) > a, \\ G_0(\mathrm{RL}(\boldsymbol{C},\boldsymbol{h}^{(k+1)})) > G_0(\mathrm{RL}(\boldsymbol{C},\boldsymbol{h}^{(k)})), & \text{otherwise.} \end{cases}$$

Therefore, the algorithm is a valid bisection search for $G_0(\mathrm{RL}(\boldsymbol{C},\boldsymbol{h}))$, and the final solution $\boldsymbol{h}^*$ satisfies $G_0(\mathrm{RL}(\boldsymbol{C},\boldsymbol{h}^*)) = a$. Furthermore, at the kth iteration for $k = 1, 2, \ldots$, the constraint $G_0(\mathrm{RL}_j(\boldsymbol{C}, h_j^{(k)})) = G_0(\mathrm{RL}_1(\boldsymbol{C}, h_1^{(k)}))$ is satisfied for all $j = 2, \ldots, J$, since the inner loop of Algorithm 2 applies a bisection search on $G_0(\mathrm{RL}_j(\boldsymbol{C}, h_j))$, which is a non-decreasing function of h_j. Since the constraint is satisfied for all $k = 1, 2, \ldots$, it will also be satisfied in $\boldsymbol{h}^*$.

□

References

Bizuneh, B., Wang, F.-K.: Comparison of different control charts for a Weibull process with type-I censoring. Communications in Statistics - Simulation and Computation **48**(4), 1088–1100 (2019) https://doi.org/10.1080/03610918.2017.1406508

Capizzi, G., Masarotto, G.: Practical Design of Generalized Likelihood Ratio Control Charts for Autocorrelated Data. Technometrics **50**(3), 357–370 (2008) https://doi.org/10.1198/004017008000000280

Capizzi, G., Masarotto, G.: Bootstrap-Based Design of Residual Control Charts. IIE Transactions **41**(4), 275–286 (2009) https://doi.org/10.1080/07408170802120059

Capizzi, G., Masarotto, G.: Efficient control chart calibration by simulated stochastic approximation. IIE Transactions **48**(1), 57–65 (2016) https://doi.org/10.1080/0740817X.2015.1055392

Chatterjee, S., Qiu, P.: Distribution-Free Cumulative Sum Control Charts Using Bootstrap-Based Control Limits. The Annals of Applied Statistics **3**(1), 349–369 (2009) https://doi.org/10.1214/08-AOAS197

Crosier, R.B.: A New Two-Sided Cumulative Sum Quality Control Scheme. Technometrics **28**(3), 187–194 (1986) https://doi.org/10.1080/00401706.1986.10488126

Crosier, R.B.: Multivariate Generalizations of Cumulative Sum Quality-Control Schemes. Technometrics **30**(3), 291–303 (1988) https://doi.org/10.2307/1270083 1270083

Crowder, S.V.: Design of Exponentially Weighted Moving Average Schemes. Journal of Quality Technology **21**(3), 155–162 (1989) https://doi.org/10.1080/00224065.1989.11979164

Dickinson, R.M., Roberts, D.A.O., Driscoll, A.R., Woodall, W.H., Vining, G.G.: CUSUM Charts for Monitoring the Characteristic Life of Censored Weibull Lifetimes. Journal of Quality Technology **46**(4), 340–358 (2014) https://doi.org/10.1080/00224065.2014.11917976

Gan, F.F.: An Optimal Design of EWMA Control Charts Based on Median Run Length. Journal of Statistical Computation and Simulation **45**(3-4), 169–184 (1993) https://doi.org/10.1080/00949659308811479

Gan, F.F.: An Optimal Design of Cumulative Sum Control Chart Based on Median Run Length. Communications in Statistics - Simulation and Computation **23**(2), 485–503 (1994) https://doi.org/10.1080/03610919408813183

Gan, F.F.: Joint Monitoring of Process Mean and Variance Using Exponentially Weighted Moving Average Control Charts. Technometrics **37**(4), 446–453 (1995) https://doi.org/10.2307/1269736 1269736

Gandy, A., Kvaløy, J.T.: Guaranteed Conditional Performance of Control Charts via Bootstrap Methods. Scandinavian Journal of Statistics **40**(4), 647–668 (2013) https://doi.org/10.1002/sjos.12006

Graham, M.A., Mukherjee, A., Chakraborti, S.: Design and Implementation Issues for a Class of Distribution-Free Phase II EWMA Exceedance Control Charts. International Journal of Production Research **55**(8), 2397–2430 (2017) https://doi.org/10.1080/00207543.2016.1249428

Hu, X., Castagliola, P., Tang, A., Zhong, J.: Guaranteed Conditional Performance of the median run length based ewma x chart with Unknown Process Parameters. Communications in Statistics - Simulation and Computation **50**(12), 4280–4299 (2021) https://doi.org/10.1080/03610918.2019.1642485

Han, D., Tsung, F.: Detection and diagnosis of unknown abrupt changes using cusum multi-chart schemes. Sequential Analysis **26**(3), 225–249 (2007) https://doi.org/10.1080/07474940701404765

Han, D., Tsung, F., Hu, X., Wang, K.: CUSUM and EWMA Multi-Charts for Detecting a Range of Mean Shifts. Statistica Sinica **17**(3), 1139–1164 (2007) 24307716

Huang, Q., Zhou, S., Shi, J.: Diagnosis of Multi-Operational Machining Processes Through Variation Propagation Analysis. Robotics and Computer-Integrated Manufacturing **18**(3), 233–239 (2002) https://doi.org/10.1016/S0736-5845(02)00014-5

Jin, J., Shi, J.: State Space Modeling of Sheet Metal Assembly for Dimensional Control. Journal of Manufacturing Science and Engineering **121**(4), 756–762 (1999) https://doi.org/10.1115/1.2833137

Knoth, S., Morais, M.C.: On ARL-Unbiased Control Charts. In: Knoth, S., Schmid, W. (eds.) Frontiers in Statistical Quality Control 11. Frontiers in Statistical Quality Control, pp. 95–117. Springer International Publishing, Cham (2015). https://doi.org/10.1007/978-3-319-12355-4_7

Knoth, S.: Run Length Quantiles of EWMA Control Charts Monitoring Normal Mean or/and Variance. International Journal of Production Research **53**(15), 4629–4647 (2015) https://doi.org/10.1080/00207543.2015.1005253

Kushner, H., Yin, G.G.: Stochastic Approximation and Recursive Algorithms and Applications, 2nd edn. Springer, New York (2003)

Lai, X., Lian, X., Liu, L., Wang, J., Liu, Y., Chong, K.C.: Generalized likelihood ratio based risk-adjusted control chart for zero-inflated Poisson process. Quality and Reliability Engineering International **39**(1), 363–381 (2023) https://doi.org/10.

1002/qre.3244

Liu, K., Mei, Y., Shi, J.: An Adaptive Sampling Strategy for Online High-Dimensional Process Monitoring. Technometrics **57**(3), 305–319 (2015) https://doi.org/10.1080/00401706.2014.947005

Li, Z., Zou, C., Gong, Z., Wang, Z.: The Computation of Average Run Length and Average Time to Signal: An Overview. Journal of Statistical Computation and Simulation **84**(8), 1779–1802 (2014) https://doi.org/10.1080/00949655.2013.766737

Montgomery, D.C., Mastrangelo, C.M.: Some Statistical Process Control Methods for Autocorrelated Data. Journal of Quality Technology **23**(3), 179–193 (1991) https://doi.org/10.1080/00224065.1991.11979321

Montgomery, D.C.: Introduction to Statistical Quality Control, 8th edn. Wiley, Hoboken, NJ (2020)

Mason, R.L., Young, J.C.: Multivariate Statistical Process Control with Industrial Applications. ASA-SIAM Series on Statistics and Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, PA (2002). https://doi.org/10.1137/1.9780898718461

Mahmoud, M.A., Zahran, A.R.: A Multivariate Adaptive Exponentially Weighted Moving Average Control Chart. Communications in Statistics - Theory and Methods **39**(4), 606–625 (2010) https://doi.org/10.1080/03610920902755813

Page, E.S.: Continuous Inspection Schemes. Biometrika **41**(1/2), 100–115 (1954) https://doi.org/10.2307/2333009 2333009

Qiu, P.: Distribution-Free Multivariate Process Control Based on Log-Linear Modeling. IIE Transactions **40**(7), 664–677 (2008) https://doi.org/10.1080/07408170701744843

Qiu, P.: Introduction to Statistical Process Control. CRC Press, Boca Raton, FL (2013)

Qiao, Y., Sun, J., Castagliola, P., Hu, X.: Optimal Design of One-Sided Exponential EWMA Charts Based on Median Run Length and Expected Median Run Length. Communications in Statistics - Theory and Methods **51**(9), 2887–2907 (2022) https://doi.org/10.1080/03610926.2020.1782937

Qiu, P., Xie, X.: Transparent Sequential Learning for Statistical Process Control of Serially Correlated Data. Technometrics **64**(4), 487–501 (2022) https://doi.org/10.1080/00401706.2021.1929493

Qiu, P., You, L.: Dynamic Disease Screening by Joint Modelling of Survival and Longitudinal Data. Journal of the Royal Statistical Society: Series C (Applied Statistics) **71**, 1158–1180 (2022) https://doi.org/10.1111/rssc.12573

Robbins, H., Monro, S.: A Stochastic Approximation Method. The Annals of Mathematical Statistics **22**(3), 400–407 (1951) https://doi.org/10.1214/aoms/1177729586

Roberts, S.W.: Control Chart Tests Based on Geometric Moving Averages. Technometrics **1**(3), 239–250 (1959) https://doi.org/10.1080/00401706.1959.10489860

Reynolds, M.R., Stoumbos, Z.G.: Combinations of Multivariate Shewhart and MEWMA Control Charts for Monitoring the Mean Vector and Covariance Matrix. Journal of Quality Technology **40**(4), 381–393 (2008) https://doi.org/10.1080/00224065.2008.11917744

Shewhart, W.A.: Economic Control of Quality Of Manufactured Product. D. Van Nostrand Company, New York, NY (1931)

Waldmann, K.-H.: Bounds for the Distribution of the Run Length of Geometric Moving Average Charts. Journal of the Royal Statistical Society. Series C (Applied Statistics) **35**(2), 151–158 (1986) https://doi.org/10.2307/2347265 2347265

Waldmann, K.-H.: Bounds for the Distribution of the Run Length of One-Sided and Two-Sided CUSUM Quality Control Schemes. Technometrics **28**(1), 61–67 (1986) https://doi.org/10.2307/1269604 1269604

Xie, X., Qiu, P.: Control Charts for Dynamic Process Monitoring with an Application to Air Pollution Surveillance. The Annals of Applied Statistics **17**(1), 47–66 (2023) https://doi.org/10.1214/22-AOAS1615

Xie, X., Qiu, P.: A General Framework for Robust Monitoring of Multivariate Correlated Processes. Technometrics, 1–22 (2023) https://doi.org/10.1080/00401706.2023.2224411

Xian, X., Wang, A., Liu, K.: A Nonparametric Adaptive Sampling Strategy for Online Monitoring of Big Data Streams. Technometrics **60**(1), 14–25 (2018) https://doi.org/10.1080/00401706.2017.1317291

Yang, K., Qiu, P.: Online sequential monitoring of spatio-temporal disease incidence rates. IISE Transactions **52**, 1218–1233 (2020)

Ye, H., Xian, X., Cheng, J.-R.C., Hable, B., Shannon, R.W., Elyaderani, M.K., Liu, K.: Online Nonparametric Monitoring of Heterogeneous Data Streams with Partial Observations Based on Thompson Sampling **55**(4), 392–404 (2023) https://doi.org/10.1080/24725854.2022.2039423

Zhou, S., Huang, Q., Shi, J.: State Space Modeling of Dimensional Variation Propagation in Multistage Machining Process Using Differential Motion Vectors. IEEE Transactions on Robotics and Automation **19**(2), 296–309 (2003) https://doi.org/

10.1109/TRA.2003.808852