Università degli Studi di Padova

Optimal Constrained Design of Control Charts Using Stochastic Approximations

Daniele Zago[1] Giovanna Capizzi[1] Peihua Qiu[2]

2023 INFORMS Annual Meeting October 15, 2023

[1] University of Padova, Padova, Italy
[2] University of Florida, Gainesville, USA

Introduction on Statistical Process Monitoring

Statistical Process Monitoring (SPM) – General framework

Framework

- The goal of SPM is to detect the presence of a **change** in a **sequential process**

$$X_n \sim \begin{cases} F_0(\cdot) & \text{for } n = 0, 1, 2, \ldots, \tau - 1 \quad \text{IN-CONTROL (IC)}, \\ F_1(\cdot) & \text{for } n = \tau, \tau + 1, 2, \ldots \quad \text{OUT-OF-CONTROL (OC)}. \end{cases}$$

- Control charts are the main tools to test the stability of the process using incoming observations.

Control charts

1. Monitoring statistic: C_n, e.g. $C_n = \max\left\{0, C_{n-1} + \left(\frac{X_n - \mu_0}{\sigma_0}\right) - k\right\}$.
2. Control limit: $h > 0$ for all $n > 0$.
3. **Run length**: $\mathrm{RL} = \inf\{n : C_n > h\}$.

Statistical Process Monitoring (SPM) – General framework

Framework

- The goal of SPM is to detect the presence of a **change** in a **sequential process**

$$X_n \sim \begin{cases} F_0(\cdot) & \text{for } n = 0, 1, 2, \ldots, \tau - 1 \quad \text{IN-CONTROL (IC)}, \\ F_1(\cdot) & \text{for } n = \tau, \tau + 1, 2, \ldots \quad \text{OUT-OF-CONTROL (OC)}. \end{cases}$$

- Control charts are the main tools to test the stability of the process using incoming observations.

Control charts

1. Monitoring statistic: C_n, e.g. $C_n = \max\left\{0, C_{n-1} + \left(\frac{X_n - \mu_0}{\sigma_0}\right) - k\right\}$.
2. Control limit: $h > 0$ for all $n > 0$.
3. **Run length**: $\mathrm{RL} = \inf\{n : C_n > h\}$.

Statistical Process Monitoring (SPM) – General framework

Framework

- The goal of SPM is to detect the presence of a **change** in a **sequential process**

$$X_n \sim \begin{cases} F_0(\cdot) & \text{for } n = 0, 1, 2, \ldots, \tau - 1 \quad \text{IN-CONTROL (IC)}, \\ F_1(\cdot) & \text{for } n = \tau, \tau + 1, 2, \ldots \quad \text{OUT-OF-CONTROL (OC)}. \end{cases}$$

- Control charts are the main tools to test the stability of the process using incoming observations.

Control charts

1. Monitoring statistic: C_n, e.g. $C_n = \max\left\{0, C_{n-1} + \left(\frac{X_n - \mu_0}{\sigma_0}\right) - k\right\}$.
2. Control limit: $h > 0$ for all $n > 0$.
3. **Run length**: $\mathrm{RL} = \inf\{n : C_n > h\}$.

Statistical Process Monitoring (SPM) – General framework

Framework

- The goal of SPM is to detect the presence of a **change** in a **sequential process**

$$X_n \sim \begin{cases} F_0(\cdot) & \text{for } n = 0, 1, 2, \ldots, \tau - 1 \quad \text{IN-CONTROL (IC)}, \\ F_1(\cdot) & \text{for } n = \tau, \tau + 1, 2, \ldots \quad \text{OUT-OF-CONTROL (OC)}. \end{cases}$$

- Control charts are the main tools to test the stability of the process using incoming observations.

Control charts

1. Monitoring statistic: C_n, e.g. $C_n = \max\left\{0, C_{n-1} + \left(\frac{X_n - \mu_0}{\sigma_0}\right) - k\right\}$.
2. Control limit: $h > 0$ for all $n > 0$.
3. **Run length**: $\mathrm{RL} = \inf\{n : C_n > h\}$.

Statistical Process Monitoring (SPM) – General framework

Framework

- The goal of SPM is to detect the presence of a **change** in a **sequential process**

$$X_n \sim \begin{cases} F_0(\cdot) & \text{for } n = 0, 1, 2, \ldots, \tau - 1 \quad \text{IN-CONTROL (IC)}, \\ F_1(\cdot) & \text{for } n = \tau, \tau + 1, 2, \ldots \quad \text{OUT-OF-CONTROL (OC)}. \end{cases}$$

- Control charts are the main tools to test the stability of the process using incoming observations.

Control charts

1. Monitoring statistic: C_n, e.g. $C_n = \max\left\{0, C_{n-1} + \left(\frac{X_n - \mu_0}{\sigma_0}\right) - k\right\}$.
2. Control limit: $h > 0$ for all $n > 0$.
3. **Run length**: $\mathrm{RL} = \inf\{n : C_n > h\}$.

Statistical Process Monitoring (SPM) – General framework

Design of a control chart

- Selection of control chart tuning parameters, which typically depend on the expected process shift.
- Selection of the control limit h, which is usually chosen so that

$$\mathrm{ARL}_{\mathrm{IC}} = \mathbb{E}[\mathrm{RL}|\tau = \infty] = \mathrm{ARL}_0,$$

  for some value of ARL_0 (e.g. 200, 370, 500, ...)

Performance metrics

- Small values of $\mathrm{ARL}_{\tau_0} = \mathbb{E}[\mathrm{RL}|\tau = \tau_0]$ mean that the chart performs better.

Statistical Process Monitoring (SPM) – General framework

Design of a control chart

- Selection of control chart tuning parameters, which typically depend on the expected process shift.
- Selection of the control limit h, which is usually chosen so that

$$\mathrm{ARL}_{\mathrm{IC}} = \mathbb{E}[\mathrm{RL}|\tau = \infty] = \mathrm{ARL}_0,$$

  for some value of ARL_0 (e.g. 200, 370, 500, ...)

Performance metrics

- Small values of $\mathrm{ARL}_{\tau_0} = \mathbb{E}[\mathrm{RL}|\tau = \tau_0]$ mean that the chart performs better.

Statistical Process Monitoring (SPM) – General framework

Design of a control chart

- Selection of control chart tuning parameters, which typically depend on the expected process shift.
- Selection of the control limit h, which is usually chosen so that

$$\mathrm{ARL}_{\mathrm{IC}} = \mathbb{E}[\mathrm{RL}|\tau = \infty] = \mathrm{ARL}_0,$$

  for some value of ARL_0 (e.g. 200, 370, 500, ...)

Performance metrics

- Small values of $\mathrm{ARL}_{\tau_0} = \mathbb{E}[\mathrm{RL}|\tau = \tau_0]$ mean that the chart performs better.

Example of (analytical) optimization

- Consider a CUSUM control chart $C_n = \max\left\{0, C_{n-1} + \left(\frac{X_n - \mu_0}{\sigma_0}\right) - k\right\}$ for monitoring changes in the mean of the process.
- Assume that the IC process observations are i.i.d. $X_n \sim N(\mu_0, \sigma_0)$.
- Suppose that the expected OC mean is μ_1 and the (standardized) change to be detected is $\delta = \frac{\mu_1 - \mu_0}{\sigma_0}$.
- Then, it is well-known that the tuning parameter that minimizes the ARL_1 is $k = \delta/2$, irrespective of the value of ARL_0.

Example of (analytical) optimization

- Consider a CUSUM control chart $C_n = \max\left\{0, C_{n-1} + \left(\frac{X_n - \mu_0}{\sigma_0}\right) - k\right\}$ for monitoring changes in the mean of the process.
- Assume that the IC process observations are i.i.d. $X_n \sim \mathcal{N}(\mu_0, \sigma_0)$.
- Suppose that the expected OC mean is μ_1 and the (standardized) change to be detected is $\delta = \frac{\mu_1 - \mu_0}{\sigma_0}$.
- Then, it is well-known that the tuning parameter that minimizes the ARL_1 is $k = \delta/2$, irrespective of the value of ARL_0.

Example of (analytical) optimization

- Consider a CUSUM control chart $C_n = \max\left\{0, C_{n-1} + \left(\frac{X_n - \mu_0}{\sigma_0}\right) - k\right\}$ for monitoring changes in the mean of the process.
- Assume that the IC process observations are i.i.d. $X_n \sim \mathcal{N}(\mu_0, \sigma_0)$.
- Suppose that the expected OC mean is μ_1 and the (standardized) change to be detected is $\delta = \frac{\mu_1 - \mu_0}{\sigma_0}$.
- Then, it is well-known that the tuning parameter that minimizes the ARL_1 is $k = \delta/2$, irrespective of the value of ARL_0.

Example of (analytical) optimization

- Consider a CUSUM control chart $C_n = \max\left\{0, C_{n-1} + \left(\frac{X_n - \mu_0}{\sigma_0}\right) - k\right\}$ for monitoring changes in the mean of the process.
- Assume that the IC process observations are i.i.d. $X_n \sim \mathcal{N}(\mu_0, \sigma_0)$.
- Suppose that the expected OC mean is μ_1 and the (standardized) change to be detected is $\delta = \frac{\mu_1 - \mu_0}{\sigma_0}$.
- Then, it is well-known that the tuning parameter that minimizes the ARL_1 is $k = \delta/2$, irrespective of the value of ARL_0.

Optimization Using Stochastic Approximations

Optimal control chart design

Chart design

- Control charts (and their run lengths) depend on a set of **tuning parameters** $\zeta \in \mathbb{R}^d$.
- Different values of tuning parameters allow better detection of different magnitudes of parameter shifts.
- The goal is to **optimize** the $\mathrm{ARL}_{\mathrm{OC}}$ under the **constraint** on the in-control ARL,

$$\zeta^* = \underset{\zeta \in \mathcal{Z}}{\operatorname{argmin}}\, \mathbb{E}_1[\mathrm{RL}(\zeta, h(\zeta))]$$
$$\text{s.t. } \mathbb{E}_0[\mathrm{RL}(\zeta, h(\zeta))] = \mathrm{ARL}_0,$$

Optimal control chart design

Chart design

- Control charts (and their run lengths) depend on a set of **tuning parameters** $\zeta \in \mathbb{R}^d$.
- Different values of tuning parameters allow better detection of different magnitudes of parameter shifts.
- The goal is to **optimize** the $\mathrm{ARL}_{\mathrm{OC}}$ under the **constraint** on the in-control ARL,

$$\zeta^* = \underset{\zeta \in \mathcal{Z}}{\operatorname{argmin}}\, \mathbb{E}_1[\mathrm{RL}(\zeta, h(\zeta))]$$
$$\text{s.t. } \mathbb{E}_0[\mathrm{RL}(\zeta, h(\zeta))] = \mathrm{ARL}_0,$$

Optimal control chart design

Chart design

- Control charts (and their run lengths) depend on a set of **tuning parameters** $\zeta \in \mathbb{R}^d$.
- Different values of tuning parameters allow better detection of different magnitudes of parameter shifts.
- The goal is to **optimize** the $\mathrm{ARL}_{\mathrm{OC}}$ under the **constraint** on the in-control ARL,

$$\zeta^* = \underset{\zeta \in \mathcal{Z}}{\operatorname{argmin}}\, \mathbb{E}_1[\mathrm{RL}(\zeta, h(\zeta))]$$
$$\text{s.t. } \mathbb{E}_0[\mathrm{RL}(\zeta, h(\zeta))] = \mathrm{ARL}_0,$$

Classical methods for optimization

Analytical methods

- Applicable only in very simple cases (CUSUM with Gaussian observations, . . .)

Numerical methods

- Numerical quadrature (for example, Knoth, 2017).
- **Limitations**: applicable for some specific control charts, scales poorly when $d > 1$.

Monte-Carlo approaches

- Estimate $\mathrm{ARL}_{\mathrm{OC}}$ for given ζ with a large number of simulations and use optimization tools.
  - Grid search (Qiu, 2008; Qiu and Xie, 2021).
  - Other numerical solvers (Capizzi and Masarotto, 2003; Mahmoud and Zahran, 2010).
- **Limitations**: function is treated as deterministic, methodologies are expensive to scale for $d > 1$ (as we will see later).

Classical methods for optimization

Analytical methods

- Applicable only in very simple cases (CUSUM with Gaussian observations, ...)

Numerical methods

- Numerical quadrature (for example, Knoth, 2017).
- **Limitations**: applicable for some specific control charts, scales poorly when $d > 1$.

Monte-Carlo approaches

- Estimate $\mathrm{ARL}_{\mathrm{OC}}$ for given ζ with a large number of simulations and use optimization tools.
  - Grid search (Qiu, 2008; Qiu and Xie, 2021).
  - Other numerical solvers (Capizzi and Masarotto, 2003; Mahmoud and Zahran, 2010).
- **Limitations**: function is treated as deterministic, methodologies are expensive to scale for $d > 1$ (as we will see later).

Classical methods for optimization

Analytical methods

- Applicable only in very simple cases (CUSUM with Gaussian observations, ...)

Numerical methods

- Numerical quadrature (for example, Knoth, 2017).
- **Limitations**: applicable for some specific control charts, scales poorly when $d > 1$.

Monte-Carlo approaches

- Estimate $\mathrm{ARL}_{\mathrm{OC}}$ for given ζ with a large number of simulations and use optimization tools.
  - Grid search (Qiu, 2008; Qiu and Xie, 2021).
  - Other numerical solvers (Capizzi and Masarotto, 2003; Mahmoud and Zahran, 2010).
- **Limitations**: function is treated as deterministic, methodologies are expensive to scale for $d > 1$ (as we will see later).

Classical methods for optimization

Analytical methods

- Applicable only in very simple cases (CUSUM with Gaussian observations, ...)

Numerical methods

- Numerical quadrature (for example, Knoth, 2017).
- **Limitations**: applicable for some specific control charts, scales poorly when $d > 1$.

Monte-Carlo approaches

- Estimate $\mathrm{ARL}_{\mathrm{OC}}$ for given ζ with a large number of simulations and use optimization tools.
  - Grid search (Qiu, 2008; Qiu and Xie, 2021).
  - Other numerical solvers (Capizzi and Masarotto, 2003; Mahmoud and Zahran, 2010).
- **Limitations**: function is treated as deterministic, methodologies are expensive to scale for $d > 1$ (as we will see later).

Classical methods for optimization

Analytical methods

- Applicable only in very simple cases (CUSUM with Gaussian observations, ...)

Numerical methods

- Numerical quadrature (for example, Knoth, 2017).
- **Limitations**: applicable for some specific control charts, scales poorly when $d > 1$.

Monte-Carlo approaches

- Estimate $\mathrm{ARL}_{\mathrm{OC}}$ for given ζ with a large number of simulations and use optimization tools.
  - Grid search (Qiu, 2008; Qiu and Xie, 2021).
  - Other numerical solvers (Capizzi and Masarotto, 2003; Mahmoud and Zahran, 2010).
- **Limitations**: function is treated as deterministic, methodologies are expensive to scale for $d > 1$ (as we will see later).

A primer on Stochastic Approximations (SA)

Starting point

- Let $Q(\zeta, h(\zeta))$ be the **noisy** function we want to minimize
- Let $\Psi : \mathbb{R}^d \to \mathcal{Z}$ be the projection onto the nearest point in $\mathcal{Z}$.
- We would like to find the minimum of Q using a gradient descent iteration (Spall, 2003)

$$\widehat{\zeta}_{k+1} = \Psi\left(\widehat{\zeta}_k - a_k \frac{\partial Q(\zeta, h(\zeta))}{\partial \zeta}\Big|_{\zeta=\widehat{\zeta}}\right), \quad k = 1, 2, \ldots \tag{1}$$

- **Problem**: The gradient in (1) is **unknown** and cannot be expressed in closed form.

Stochastic approximations approach

- Estimate the gradient using finite differences and use $\widehat{\zeta}_{k+1} = \Psi\left(\widehat{\zeta}_k - a_k \widehat{\boldsymbol{g}}_k(\widehat{\zeta}_k)\right)$,
- When $\zeta \in \mathbb{R}^d$, the SA method requires evaluating the function Q at $2d$ parameter values.
- **Problem**: Because of the $\mathrm{ARL}_{\mathrm{IC}}$ constraint, the function Q is **expensive** to evaluate.

A primer on Stochastic Approximations (SA)

Starting point

- Let $Q(\zeta, h(\zeta))$ be the **noisy** function we want to minimize
- Let $\Psi : \mathbb{R}^d \to \mathcal{Z}$ be the projection onto the nearest point in $\mathcal{Z}$.
- We would like to find the minimum of Q using a gradient descent iteration (Spall, 2003)

$$\widehat{\zeta}_{k+1} = \Psi\left(\widehat{\zeta}_k - a_k \frac{\partial Q(\zeta, h(\zeta))}{\partial \zeta}\Big|_{\zeta=\widehat{\zeta}}\right), \quad k = 1, 2, \ldots \tag{1}$$

- **Problem**: The gradient in (1) is **unknown** and cannot be expressed in closed form.

Stochastic approximations approach

- Estimate the gradient using finite differences and use $\widehat{\zeta}_{k+1} = \Psi\left(\widehat{\zeta}_k - a_k \widehat{\boldsymbol{g}}_k(\widehat{\zeta}_k)\right)$,
- When $\zeta \in \mathbb{R}^d$, the SA method requires evaluating the function Q at $2d$ parameter values.
- **Problem**: Because of the $\mathrm{ARL}_{\mathrm{IC}}$ constraint, the function Q is **expensive** to evaluate.

A primer on Stochastic Approximations (SA)

Starting point

- Let $Q(\zeta, h(\zeta))$ be the **noisy** function we want to minimize
- Let $\Psi : \mathbb{R}^d \to \mathcal{Z}$ be the projection onto the nearest point in $\mathcal{Z}$.
- We would like to find the minimum of Q using a gradient descent iteration (Spall, 2003)

$$\widehat{\zeta}_{k+1} = \Psi\left(\widehat{\zeta}_k - a_k \frac{\partial Q(\zeta, h(\zeta))}{\partial \zeta}\Big|_{\zeta=\widehat{\zeta}}\right), \quad k = 1, 2, \ldots \tag{1}$$

- **Problem**: The gradient in (1) is **unknown** and cannot be expressed in closed form.

Stochastic approximations approach

- Estimate the gradient using finite differences and use $\widehat{\zeta}_{k+1} = \Psi\left(\widehat{\zeta}_k - a_k \widehat{\boldsymbol{g}}_k(\widehat{\zeta}_k)\right)$,
- When $\zeta \in \mathbb{R}^d$, the SA method requires evaluating the function Q at $2d$ parameter values.
- **Problem**: Because of the $\mathrm{ARL}_{\mathrm{IC}}$ constraint, the function Q is **expensive** to evaluate.

A primer on Stochastic Approximations (SA)

Starting point

- Let $Q(\zeta, h(\zeta))$ be the **noisy** function we want to minimize
- Let $\Psi : \mathbb{R}^d \to \mathcal{Z}$ be the projection onto the nearest point in $\mathcal{Z}$.
- We would like to find the minimum of Q using a gradient descent iteration (Spall, 2003)

$$\widehat{\zeta}_{k+1} = \Psi\left(\widehat{\zeta}_k - a_k \frac{\partial Q(\zeta, h(\zeta))}{\partial \zeta}\Big|_{\zeta=\widehat{\zeta}}\right), \quad k = 1, 2, \ldots \tag{1}$$

- **Problem**: The gradient in (1) is **unknown** and cannot be expressed in closed form.

Stochastic approximations approach

- Estimate the gradient using finite differences and use $\widehat{\zeta}_{k+1} = \Psi\left(\widehat{\zeta}_k - a_k \widehat{\boldsymbol{g}}_k(\widehat{\zeta}_k)\right)$,
- When $\zeta \in \mathbb{R}^d$, the SA method requires evaluating the function Q at $2d$ parameter values.
- **Problem**: Because of the $\mathrm{ARL}_{\mathrm{IC}}$ constraint, the function Q is **expensive** to evaluate.

A primer on Stochastic Approximations (SA)

Starting point

- Let $Q(\zeta, h(\zeta))$ be the **noisy** function we want to minimize
- Let $\Psi : \mathbb{R}^d \to \mathcal{Z}$ be the projection onto the nearest point in $\mathcal{Z}$.
- We would like to find the minimum of Q using a gradient descent iteration (Spall, 2003)

$$\widehat{\zeta}_{k+1} = \Psi\left(\widehat{\zeta}_k - a_k \frac{\partial Q(\zeta, h(\zeta))}{\partial \zeta}\Big|_{\zeta=\widehat{\zeta}}\right), \quad k = 1, 2, \ldots \tag{1}$$

- **Problem**: The gradient in (1) is **unknown** and cannot be expressed in closed form.

Stochastic approximations approach

- Estimate the gradient using finite differences and use $\widehat{\zeta}_{k+1} = \Psi\left(\widehat{\zeta}_k - a_k \widehat{\boldsymbol{g}}_k(\widehat{\zeta}_k)\right)$,
- When $\zeta \in \mathbb{R}^d$, the SA method requires evaluating the function Q at $2d$ parameter values.
- **Problem**: Because of the $\mathrm{ARL}_{\mathrm{IC}}$ constraint, the function Q is **expensive** to evaluate.

A primer on Stochastic Approximations (SA)

Starting point

- Let $Q(\zeta, h(\zeta))$ be the **noisy** function we want to minimize
- Let $\Psi : \mathbb{R}^d \to \mathcal{Z}$ be the projection onto the nearest point in $\mathcal{Z}$.
- We would like to find the minimum of Q using a gradient descent iteration (Spall, 2003)

$$\widehat{\zeta}_{k+1} = \Psi\left(\widehat{\zeta}_k - a_k \frac{\partial Q(\zeta, h(\zeta))}{\partial \zeta}\Big|_{\zeta=\widehat{\zeta}}\right), \quad k = 1, 2, \ldots \tag{1}$$

- **Problem**: The gradient in (1) is **unknown** and cannot be expressed in closed form.

Stochastic approximations approach

- Estimate the gradient using finite differences and use $\widehat{\zeta}_{k+1} = \Psi\left(\widehat{\zeta}_k - a_k \widehat{\boldsymbol{g}}_k(\widehat{\zeta}_k)\right)$,
- When $\zeta \in \mathbb{R}^d$, the SA method requires evaluating the function Q at $2d$ parameter values.
- **Problem**: Because of the $\mathrm{ARL}_{\mathrm{IC}}$ constraint, the function Q is **expensive** to evaluate.

A primer on Stochastic Approximations (SA)

Starting point

- Let $Q(\zeta, h(\zeta))$ be the **noisy** function we want to minimize
- Let $\Psi : \mathbb{R}^d \to \mathcal{Z}$ be the projection onto the nearest point in $\mathcal{Z}$.
- We would like to find the minimum of Q using a gradient descent iteration (Spall, 2003)

$$\widehat{\zeta}_{k+1} = \Psi\left(\widehat{\zeta}_k - a_k \frac{\partial Q(\zeta, h(\zeta))}{\partial \zeta}\Big|_{\zeta=\widehat{\zeta}}\right), \quad k = 1, 2, \ldots \tag{1}$$

- **Problem**: The gradient in (1) is **unknown** and cannot be expressed in closed form.

Stochastic approximations approach

- Estimate the gradient using finite differences and use $\widehat{\zeta}_{k+1} = \Psi\left(\widehat{\zeta}_k - a_k \widehat{\boldsymbol{g}}_k(\widehat{\zeta}_k)\right)$,
- When $\zeta \in \mathbb{R}^d$, the SA method requires evaluating the function Q at $2d$ parameter values.
- **Problem**: Because of the $\mathrm{ARL}_{\mathrm{IC}}$ constraint, the function Q is **expensive** to evaluate.

Simultaneous Perturbation Stochastic Approximations (SPSA)

Simultaneous perturbations approach

- Let $\boldsymbol{\Delta}_k = (\Delta_{1k}, \ldots, \Delta_{dk})$ be independent zero-mean random variables that are symmetric.
- Typically (Spall, 2003)

$$\Delta_{jk} \overset{\text{iid}}{\sim} \begin{cases} 1 & \text{with probability } 1/2, \\ -1 & \text{with probability } 1/2. \end{cases}$$

- Perturb the current parameter estimates: $\widehat{\zeta}_k^+ = \Psi(\widehat{\zeta}_k + c_k \boldsymbol{\Delta}_k)$ and $\widehat{\zeta}_k^- = \Psi(\widehat{\zeta}_k - c_k \boldsymbol{\Delta}_k)$.
- Gradient estimate is

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

- **Advantage**: requires 2 evaluations of Q irrespective of the dimension d.

Simultaneous Perturbation Stochastic Approximations (SPSA)

Simultaneous perturbations approach

- Let $\boldsymbol{\Delta}_k = (\Delta_{1k}, \ldots, \Delta_{dk})$ be independent zero-mean random variables that are symmetric.
- Typically (Spall, 2003)

$$\Delta_{jk} \overset{\text{iid}}{\sim} \begin{cases} 1 & \text{with probability } 1/2, \\ -1 & \text{with probability } 1/2. \end{cases}$$

- Perturb the current parameter estimates: $\widehat{\zeta}_k^+ = \Psi(\widehat{\zeta}_k + c_k \boldsymbol{\Delta}_k)$ and $\widehat{\zeta}_k^- = \Psi(\widehat{\zeta}_k - c_k \boldsymbol{\Delta}_k)$.
- Gradient estimate is

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

- **Advantage**: requires 2 evaluations of Q irrespective of the dimension d.

Simultaneous Perturbation Stochastic Approximations (SPSA)

Simultaneous perturbations approach

- Let $\boldsymbol{\Delta}_k = (\Delta_{1k}, \ldots, \Delta_{dk})$ be independent zero-mean random variables that are symmetric.
- Typically (Spall, 2003)

$$\Delta_{jk} \overset{\text{iid}}{\sim} \begin{cases} 1 & \text{with probability } 1/2, \\ -1 & \text{with probability } 1/2. \end{cases}$$

- Perturb the current parameter estimates: $\widehat{\zeta}_k^+ = \Psi(\widehat{\zeta}_k + c_k \boldsymbol{\Delta}_k)$ and $\widehat{\zeta}_k^- = \Psi(\widehat{\zeta}_k - c_k \boldsymbol{\Delta}_k)$.
- Gradient estimate is

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

- **Advantage**: requires 2 evaluations of Q irrespective of the dimension d.

Simultaneous Perturbation Stochastic Approximations (SPSA)

Simultaneous perturbations approach

- Let $\boldsymbol{\Delta}_k = (\Delta_{1k}, \ldots, \Delta_{dk})$ be independent zero-mean random variables that are symmetric.
- Typically (Spall, 2003)

$$\Delta_{jk} \overset{\text{iid}}{\sim} \begin{cases} 1 & \text{with probability } 1/2, \\ -1 & \text{with probability } 1/2. \end{cases}$$

- Perturb the current parameter estimates: $\widehat{\zeta}_k^+ = \Psi(\widehat{\zeta}_k + c_k \boldsymbol{\Delta}_k)$ and $\widehat{\zeta}_k^- = \Psi(\widehat{\zeta}_k - c_k \boldsymbol{\Delta}_k)$.
- Gradient estimate is

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

- **Advantage**: requires 2 evaluations of Q irrespective of the dimension d.

Simultaneous Perturbation Stochastic Approximations (SPSA)

Simultaneous perturbations approach

- Let $\boldsymbol{\Delta}_k = (\Delta_{1k}, \ldots, \Delta_{dk})$ be independent zero-mean random variables that are symmetric.
- Typically (Spall, 2003)

$$\Delta_{jk} \overset{\text{iid}}{\sim} \begin{cases} 1 & \text{with probability } 1/2, \\ -1 & \text{with probability } 1/2. \end{cases}$$

- Perturb the current parameter estimates: $\widehat{\zeta}_k^+ = \Psi(\widehat{\zeta}_k + c_k \boldsymbol{\Delta}_k)$ and $\widehat{\zeta}_k^- = \Psi(\widehat{\zeta}_k - c_k \boldsymbol{\Delta}_k)$.
- Gradient estimate is

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

- **Advantage**: requires 2 evaluations of Q irrespective of the dimension d.

Further enhancements

Polyak averaging

- The estimate of the optimum ζ^* at iteration k is the average over the optimization path,

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

- Averaging increases stability while providing a similar convergence rate to the solution.

Noise reduction

- Once $h(\widehat{\zeta}_k)$ is found, $r = 100$ simulations of $Q(\widehat{\zeta}_k)$ are used to estimate $\widehat{\boldsymbol{g}}(\widehat{\zeta}_k)$.

Computational bottleneck

- Calculation of the control limits $h(\widehat{\zeta}_k^+)$ and $h(\widehat{\zeta}_k^-)$ is the algorithm's bottleneck.
- We use a low-precision SA algorithm (Capizzi and Masarotto, 2016) with a warm-start initialization that allows it to become more accurate as k increases.

Further enhancements

Polyak averaging

- The estimate of the optimum ζ^* at iteration k is the average over the optimization path,

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

- Averaging increases stability while providing a similar convergence rate to the solution.

Noise reduction

- Once $h(\widehat{\zeta}_k)$ is found, $r = 100$ simulations of $Q(\widehat{\zeta}_k)$ are used to estimate $\widehat{\boldsymbol{g}}(\widehat{\zeta}_k)$.

Computational bottleneck

- Calculation of the control limits $h(\widehat{\zeta}_k^+)$ and $h(\widehat{\zeta}_k^-)$ is the algorithm's bottleneck.
- We use a low-precision SA algorithm (Capizzi and Masarotto, 2016) with a warm-start initialization that allows it to become more accurate as k increases.

Further enhancements

Polyak averaging

- The estimate of the optimum ζ^* at iteration k is the average over the optimization path,

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

- Averaging increases stability while providing a similar convergence rate to the solution.

Noise reduction

- Once $h(\widehat{\zeta}_k)$ is found, $r = 100$ simulations of $Q(\widehat{\zeta}_k)$ are used to estimate $\widehat{\boldsymbol{g}}(\widehat{\zeta}_k)$.

Computational bottleneck

- Calculation of the control limits $h(\widehat{\zeta}_k^+)$ and $h(\widehat{\zeta}_k^-)$ is the algorithm's bottleneck.
- We use a low-precision SA algorithm (Capizzi and Masarotto, 2016) with a warm-start initialization that allows it to become more accurate as k increases.

Further enhancements

Polyak averaging

- The estimate of the optimum ζ^* at iteration k is the average over the optimization path,

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

- Averaging increases stability while providing a similar convergence rate to the solution.

Noise reduction

- Once $h(\widehat{\zeta}_k)$ is found, $r = 100$ simulations of $Q(\widehat{\zeta}_k)$ are used to estimate $\widehat{\boldsymbol{g}}(\widehat{\zeta}_k)$.

Computational bottleneck

- Calculation of the control limits $h(\widehat{\zeta}_k^+)$ and $h(\widehat{\zeta}_k^-)$ is the algorithm's bottleneck.
- We use a low-precision SA algorithm (Capizzi and Masarotto, 2016) with a warm-start initialization that allows it to become more accurate as k increases.

Further enhancements

Polyak averaging

- The estimate of the optimum ζ^* at iteration k is the average over the optimization path,

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

- Averaging increases stability while providing a similar convergence rate to the solution.

Noise reduction

- Once $h(\widehat{\zeta}_k)$ is found, $r = 100$ simulations of $Q(\widehat{\zeta}_k)$ are used to estimate $\widehat{\boldsymbol{g}}(\widehat{\zeta}_k)$.

Computational bottleneck

- Calculation of the control limits $h(\widehat{\zeta}_k^+)$ and $h(\widehat{\zeta}_k^-)$ is the algorithm's bottleneck.
- We use a low-precision SA algorithm (Capizzi and Masarotto, 2016) with a warm-start initialization that allows it to become more accurate as k increases.

Convergence criterion

"Stochastic" convergence criterion

- A reasonable stopping rule of the algorithm is $|\mathbb{E}[\widehat{\boldsymbol{g}}_{jk}(\widehat{\boldsymbol{\zeta}}_k)]| \leq \nu$, for a small value of ν
- We leverage the asymptotic distribution of the PR averaging scheme (Maryak, 1997),

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

- Using a similar approach to Capizzi and Masarotto (2016), a stopping criterion can be defined as

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

  - z is the $[(1-\nu)/2]$-th quantile of the standard normal distribution.
  - $N_\mathrm{m} + N_\mathrm{f}$ is specified to avoid a premature ending of the algorithm.

Convergence criterion

"Stochastic" convergence criterion

- A reasonable stopping rule of the algorithm is $|\mathbb{E}[\widehat{\boldsymbol{g}}_{jk}(\widehat{\zeta}_k)]| \leq \nu$, for a small value of ν
- We leverage the asymptotic distribution of the PR averaging scheme (Maryak, 1997),

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

- Using a similar approach to Capizzi and Masarotto (2016), a stopping criterion can be defined as

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

  - z is the $[(1-\nu)/2]$-th quantile of the standard normal distribution.
  - $N_\mathrm{m} + N_\mathrm{f}$ is specified to avoid a premature ending of the algorithm.

Convergence criterion

"Stochastic" convergence criterion

- A reasonable stopping rule of the algorithm is $|\mathbb{E}[\widehat{\boldsymbol{g}}_{jk}(\widehat{\zeta}_k)]| \leq \nu$, for a small value of ν
- We leverage the asymptotic distribution of the PR averaging scheme (Maryak, 1997),

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

- Using a similar approach to Capizzi and Masarotto (2016), a stopping criterion can be defined as

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

  - z is the $[(1-\nu)/2]$-th quantile of the standard normal distribution.
  - $N_\mathrm{m} + N_\mathrm{f}$ is specified to avoid a premature ending of the algorithm.

Convergence criterion

"Stochastic" convergence criterion

- A reasonable stopping rule of the algorithm is $|\mathbb{E}[\widehat{\boldsymbol{g}}_{jk}(\widehat{\zeta}_k)]| \leq \nu$, for a small value of ν
- We leverage the asymptotic distribution of the PR averaging scheme (Maryak, 1997),

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

- Using a similar approach to Capizzi and Masarotto (2016), a stopping criterion can be defined as

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

  - z is the $[(1-\nu)/2]$-th quantile of the standard normal distribution.
  - $N_\mathrm{m} + N_\mathrm{f}$ is specified to avoid a premature ending of the algorithm.

Convergence criterion

"Stochastic" convergence criterion

- A reasonable stopping rule of the algorithm is $|\mathbb{E}[\widehat{\boldsymbol{g}}_{jk}(\widehat{\zeta}_k)]| \leq \nu$, for a small value of ν
- We leverage the asymptotic distribution of the PR averaging scheme (Maryak, 1997),

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

- Using a similar approach to Capizzi and Masarotto (2016), a stopping criterion can be defined as

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

  - z is the $[(1-\nu)/2]$-th quantile of the standard normal distribution.
  - $N_\mathrm{m} + N_\mathrm{f}$ is specified to avoid a premature ending of the algorithm.

Convergence criterion ii

"Deterministic" convergence criterion

- The "stochastic" convergence criterion $\overline{N}_s$ is coupled with a classical "deterministic" convergence criterion,

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

Convergence criterion

- The convergence criterion used in our simulation is $\overline{N} = \min \left\{ \overline{N}_s, \overline{N}_a \right\}$.
- The convergence criteria $\overline{N}_s$ and $\overline{N}_a$ are "complementary":
  - $\overline{N}_s$ is useful for optimizing "flat" functions, where the variance is small.
  - $\overline{N}_a$ can help when functions have high curvature, because $\widehat{\zeta}_k$ will "jump around" the optimum and the effect will be averaged out in $\bar{\zeta}_k$ (Maryak, 1997).

Convergence criterion ii

"Deterministic" convergence criterion

- The "stochastic" convergence criterion $\overline{N}_s$ is coupled with a classical "deterministic" convergence criterion,

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

Convergence criterion

- The convergence criterion used in our simulation is $\overline{N} = \min \left\{ \overline{N}_s, \overline{N}_a \right\}$.
- The convergence criteria $\overline{N}_s$ and $\overline{N}_a$ are "complementary":
  - $\overline{N}_s$ is useful for optimizing "flat" functions, where the variance is small.
  - $\overline{N}_a$ can help when functions have high curvature, because $\widehat{\zeta}_k$ will "jump around" the optimum and the effect will be averaged out in $\bar{\zeta}_k$ (Maryak, 1997).

Convergence criterion ii

"Deterministic" convergence criterion

- The "stochastic" convergence criterion $\overline{N}_s$ is coupled with a classical "deterministic" convergence criterion,

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

Convergence criterion

- The convergence criterion used in our simulation is $\overline{N} = \min \left\{ \overline{N}_s, \overline{N}_a \right\}$.
- The convergence criteria $\overline{N}_s$ and $\overline{N}_a$ are "complementary":
  - $\overline{N}_s$ is useful for optimizing "flat" functions, where the variance is small.
  - $\overline{N}_a$ can help when functions have high curvature, because $\widehat{\zeta}_k$ will "jump around" the optimum and the effect will be averaged out in $\bar{\zeta}_k$ (Maryak, 1997).

Convergence criterion ii

"Deterministic" convergence criterion

- The "stochastic" convergence criterion $\overline{N}_s$ is coupled with a classical "deterministic" convergence criterion,

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

Convergence criterion

- The convergence criterion used in our simulation is $\overline{N} = \min \left\{ \overline{N}_s, \overline{N}_a \right\}$.
- The convergence criteria $\overline{N}_s$ and $\overline{N}_a$ are "complementary":
  - $\overline{N}_s$ is useful for optimizing "flat" functions, where the variance is small.
  - $\overline{N}_a$ can help when functions have high curvature, because $\widehat{\zeta}_k$ will "jump around" the optimum and the effect will be averaged out in $\bar{\zeta}_k$ (Maryak, 1997).

Convergence criterion ii

"Deterministic" convergence criterion

- The "stochastic" convergence criterion $\overline{N}_s$ is coupled with a classical "deterministic" convergence criterion,

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

Convergence criterion

- The convergence criterion used in our simulation is $\overline{N} = \min \left\{ \overline{N}_s, \overline{N}_a \right\}$.
- The convergence criteria $\overline{N}_s$ and $\overline{N}_a$ are "complementary":
  - $\overline{N}_s$ is useful for optimizing "flat" functions, where the variance is small.
  - $\overline{N}_a$ can help when functions have high curvature, because $\widehat{\zeta}_k$ will "jump around" the optimum and the effect will be averaged out in $\bar{\zeta}_k$ (Maryak, 1997).

Numerical results

Comparison with traditional methods

Grid search

- At iteration k, divide each coordinate of a grid $[\zeta_{\min}^{(k)}, \zeta_{\max}^{(k)}]$ in m segments and calculate the objective function at the endpoints.
- Find the endpoint $\zeta^{(k+1)}$ with minimum value of the objective.
- The endpoints adjacent to $\zeta^{(k+1)}$ define $[\zeta_{\min}^{(k+1)}, \zeta_{\max}^{(k+1)}]$.

Nelder-Mead

- Method based on reflection, extension, contraction, and shrinkage of a simplex.
- An efficient implementation is available in the `NLopt.jl` package.

Comparison with traditional methods

Grid search

- At iteration k, divide each coordinate of a grid $[\zeta_{\min}^{(k)}, \zeta_{\max}^{(k)}]$ in m segments and calculate the objective function at the endpoints.
- Find the endpoint $\zeta^{(k+1)}$ with minimum value of the objective.
- The endpoints adjacent to $\zeta^{(k+1)}$ define $[\zeta_{\min}^{(k+1)}, \zeta_{\max}^{(k+1)}]$.

Nelder-Mead

- Method based on reflection, extension, contraction, and shrinkage of a simplex.
- An efficient implementation is available in the `NLopt.jl` package.

Comparison with traditional methods

Grid search

- At iteration k, divide each coordinate of a grid $[\zeta_{\min}^{(k)}, \zeta_{\max}^{(k)}]$ in m segments and calculate the objective function at the endpoints.
- Find the endpoint $\zeta^{(k+1)}$ with minimum value of the objective.
- The endpoints adjacent to $\zeta^{(k+1)}$ define $[\zeta_{\min}^{(k+1)}, \zeta_{\max}^{(k+1)}]$.

Nelder-Mead

- Method based on reflection, extension, contraction, and shrinkage of a simplex.
- An efficient implementation is available in the `NLopt.jl` package.

Comparison with traditional methods

Grid search

- At iteration k, divide each coordinate of a grid $[\zeta_{\min}^{(k)}, \zeta_{\max}^{(k)}]$ in m segments and calculate the objective function at the endpoints.
- Find the endpoint $\zeta^{(k+1)}$ with minimum value of the objective.
- The endpoints adjacent to $\zeta^{(k+1)}$ define $[\zeta_{\min}^{(k+1)}, \zeta_{\max}^{(k+1)}]$.

Nelder-Mead

- Method based on reflection, extension, contraction, and shrinkage of a simplex.
- An efficient implementation is available in the `NLopt.jl` package.

Comparison with traditional methods

Grid search

- At iteration k, divide each coordinate of a grid $[\zeta_{\min}^{(k)}, \zeta_{\max}^{(k)}]$ in m segments and calculate the objective function at the endpoints.
- Find the endpoint $\zeta^{(k+1)}$ with minimum value of the objective.
- The endpoints adjacent to $\zeta^{(k+1)}$ define $[\zeta_{\min}^{(k+1)}, \zeta_{\max}^{(k+1)}]$.

Nelder-Mead

- Method based on reflection, extension, contraction, and shrinkage of a simplex.
- An efficient implementation is available in the `NLopt.jl` package.

Results

- CUSUM control chart: $C_t = \max\{0, C_{t-1} + X_t - k\}$, $X_t \sim N(\delta, 1)$ for various δ's.
- Optimal parameter to detect the mean shift δ is $k^* = \delta/2$.

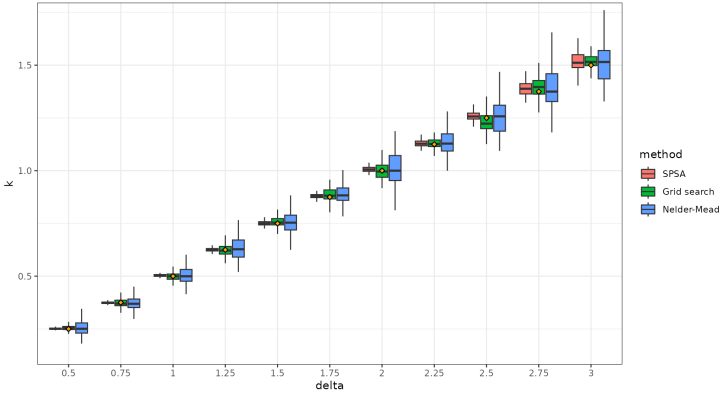

Figure 1: Estimated optimal parameter values over 100 optimizations.

Results

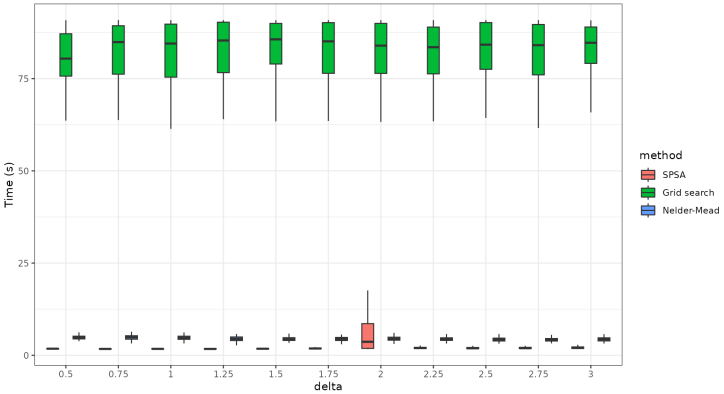

Figure 2: Computing times over 100 optimization.

Multidimensional scalability

- MEWMA control chart: $C_t = (I - \Lambda)C_{t-1} + \Lambda X_t$, with $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_d)$ and $\lambda_j \in (0, 1)$ for all j. $X_{t,1} \sim \mathcal{N}(\delta, 1), X_{t,2} \sim \chi^2_{1+\delta\sqrt{2}}, X_{t,3} \sim \text{Pois}(1 + \delta)$.

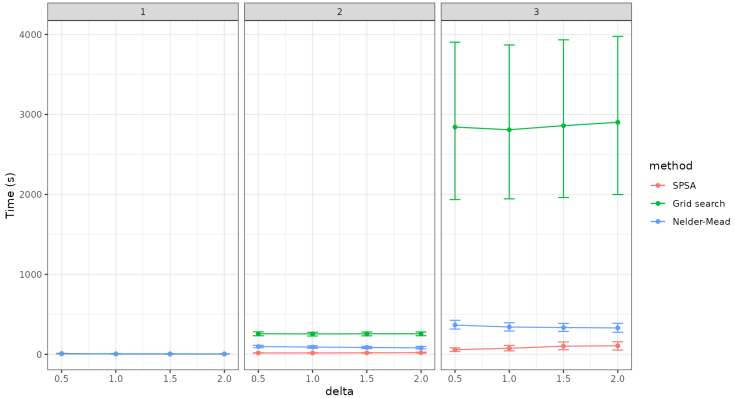

Figure 3: Median, 0.1th and 0.9th quantiles of the computing times over 100 optimizations.

Extension to other metrics

Generalizations

- Sometimes, other metrics such as the median run length or quantiles of the run length are of interest (Knoth, 2015).
- The proposed SPSA algorithm is flexible enough to be generalized to other performance metrics based on the RL.

Example: median run length

- Minimization of the out-of-control median run length with a constraint on the in-control median run length.
- Q becomes the median of the r simulated out-of-control run lengths.
- The control limit is estimated using a modified gradient iteration in the SA algorithm (Capizzi and Masarotto, 2009)

Extension to other metrics

Generalizations

- Sometimes, other metrics such as the median run length or quantiles of the run length are of interest (Knoth, 2015).
- The proposed SPSA algorithm is flexible enough to be generalized to other performance metrics based on the RL.

Example: median run length

- Minimization of the out-of-control median run length with a constraint on the in-control median run length.
- Q becomes the median of the r simulated out-of-control run lengths.
- The control limit is estimated using a modified gradient iteration in the SA algorithm (Capizzi and Masarotto, 2009)

Extension to other metrics

Generalizations

- Sometimes, other metrics such as the median run length or quantiles of the run length are of interest (Knoth, 2015).
- The proposed SPSA algorithm is flexible enough to be generalized to other performance metrics based on the RL.

Example: median run length

- Minimization of the out-of-control median run length with a constraint on the in-control median run length.
- Q becomes the median of the r simulated out-of-control run lengths.
- The control limit is estimated using a modified gradient iteration in the SA algorithm (Capizzi and Masarotto, 2009)

Extension to other metrics

Generalizations

- Sometimes, other metrics such as the median run length or quantiles of the run length are of interest (Knoth, 2015).
- The proposed SPSA algorithm is flexible enough to be generalized to other performance metrics based on the RL.

Example: median run length

- Minimization of the out-of-control median run length with a constraint on the in-control median run length.
- Q becomes the median of the r simulated out-of-control run lengths.
- The control limit is estimated using a modified gradient iteration in the SA algorithm (Capizzi and Masarotto, 2009)

Extension to other metrics

Generalizations

- Sometimes, other metrics such as the median run length or quantiles of the run length are of interest (Knoth, 2015).
- The proposed SPSA algorithm is flexible enough to be generalized to other performance metrics based on the RL.

Example: median run length

- Minimization of the out-of-control median run length with a constraint on the in-control median run length.
- Q becomes the median of the r simulated out-of-control run lengths.
- The control limit is estimated using a modified gradient iteration in the SA algorithm (Capizzi and Masarotto, 2009)

Conclusion

Summary

- We have proposed a novel methodology for a more **efficient design** of control charts tuning parameters.
- The only requirement is to be able to **simulate run lengths** from the IC process (e.g. using parametric/nonparametric bootstrap, bootstrap for time series, ...)
- The methodology is based on a **stochastic approximations** algorithm that is specifically designed for the constrained optimization problem.
- Numerical simulation suggest that the approach is **more efficient** than traditional approaches, especially for **multi-dimensional** tuning parameters.
- Due to its flexibility, it can be **generalized** to the optimization of various performance metrics such as median run length, run length quantiles, etc.

Summary

- We have proposed a novel methodology for a more **efficient design** of control charts tuning parameters.
- The only requirement is to be able to **simulate run lengths** from the IC process (e.g. using parametric/nonparametric bootstrap, bootstrap for time series, ...)
- The methodology is based on a **stochastic approximations** algorithm that is specifically designed for the constrained optimization problem.
- Numerical simulation suggest that the approach is **more efficient** than traditional approaches, especially for **multi-dimensional** tuning parameters.
- Due to its flexibility, it can be **generalized** to the optimization of various performance metrics such as median run length, run length quantiles, etc.

Summary

- We have proposed a novel methodology for a more **efficient design** of control charts tuning parameters.
- The only requirement is to be able to **simulate run lengths** from the IC process (e.g. using parametric/nonparametric bootstrap, bootstrap for time series, ...)
- The methodology is based on a **stochastic approximations** algorithm that is specifically designed for the constrained optimization problem.
- Numerical simulation suggest that the approach is **more efficient** than traditional approaches, especially for **multi-dimensional** tuning parameters.
- Due to its flexibility, it can be **generalized** to the optimization of various performance metrics such as median run length, run length quantiles, etc.

Summary

- We have proposed a novel methodology for a more **efficient design** of control charts tuning parameters.
- The only requirement is to be able to **simulate run lengths** from the IC process (e.g. using parametric/nonparametric bootstrap, bootstrap for time series, ...)
- The methodology is based on a **stochastic approximations** algorithm that is specifically designed for the constrained optimization problem.
- Numerical simulation suggest that the approach is **more efficient** than traditional approaches, especially for **multi-dimensional** tuning parameters.
- Due to its flexibility, it can be **generalized** to the optimization of various performance metrics such as median run length, run length quantiles, etc.

Summary

- We have proposed a novel methodology for a more **efficient design** of control charts tuning parameters.
- The only requirement is to be able to **simulate run lengths** from the IC process (e.g. using parametric/nonparametric bootstrap, bootstrap for time series, ...)
- The methodology is based on a **stochastic approximations** algorithm that is specifically designed for the constrained optimization problem.
- Numerical simulation suggest that the approach is **more efficient** than traditional approaches, especially for **multi-dimensional** tuning parameters.
- Due to its flexibility, it can be **generalized** to the optimization of various performance metrics such as median run length, run length quantiles, etc.

Thank you for the attention

Questions

References i

Capizzi, G. and Masarotto, G. (2003). “An Adaptive Exponentially Weighted Moving Average Control Chart”. In: *Technometrics* 45.3, 199–207.

Capizzi, G. and Masarotto, G. (2009). “Bootstrap-Based Design of Residual Control Charts”. In: *IIE Transactions* 41.4, 275–286.

Capizzi, G. and Masarotto, G. (2016). “Efficient Control Chart Calibration by Simulated Stochastic Approximation”. In: *IIE Transactions* 48.1, 57–65.

Knoth, S. (2015). “Run Length Quantiles of EWMA Control Charts Monitoring Normal Mean or/and Variance”. In: *International Journal of Production Research* 53.15, 4629–4647.

Knoth, S. (2017). “ARL Numerics for MEWMA Charts”. In: *Journal of Quality Technology* 49.1, 78–89.

References ii

Mahmoud, M. A. and Zahran, A. R. (2010). “A Multivariate Adaptive Exponentially Weighted Moving Average Control Chart”. In: *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*. Vol. 3, 2287–2290 vol.3.

Qiu, P. (2008). “Distribution-Free Multivariate Process Control Based on Log-Linear Modeling”. In: *IIE Transactions* 40.7, 664–677.

Qiu, P. and Xie, X. (2021). “Transparent Sequential Learning for Statistical Process Control of Serially Correlated Data”. In: *Technometrics* 0.0, 1–15.

References iii

Spall, J. (1992). “Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation”. In: *IEEE Transactions on Automatic Control* 37.3, 332–341.

Spall, J. (1998). “Implementation of the Simultaneous Perturbation Algorithm for Stochastic Optimization”. In: *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. Hoboken, N.J: Wiley-Interscience.

Selection of the tuning constants in the SPSA algorithm i

- The proposed SPSA algorithm requires the selection of many tuning constants.
- A semi-automated way of selecting most of the tuning constants is available following the guidelines by Spall (1992), Spall (1998) and Spall (2003).
  - The gain sequences in the algorithm are 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 $\widehat{\zeta}_k$ to ζ^* as $k \longrightarrow \infty$ under some quite general assumptions, as proved by Spall (1992).
  - The constant a, A and c require a small preliminary adaptive step in order to be estimated.
  - c can be approximately set to be the standard deviation $\sigma_{\widehat{\zeta}_0}$ of the OC RL calculated at the initial value $\widehat{\zeta}_0$ (Spall, 1998).
  - In numerical studies, we have seen that setting $c = \min\{\widehat{\sigma}_{\widehat{\zeta}_0}, 0.1\}$ can avoid excessive perturbation of the tuning parameters in the early iterations.
  - 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 $A = 0.1 \times 150 = 15$.

Selection of the tuning constants in the SPSA algorithm ii

- (Spall, 1998) recommends selecting a to be the expected magnitude change in $\widehat{\zeta}_k$ during the first few iterations. Specifically,

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

  where s is the initial step size and $\overline{G} = \frac{1}{d} \sum_{j=1}^{d} \sum_{l=1}^{n_c} \widehat{\boldsymbol{g}}_{jl}(\widehat{\zeta}_0)/n_c$ is a preliminary estimate of the average value of the gradient in $\widehat{\zeta}_0$ 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.