



# MPS-APO: a rapid and automatic parameter optimizer for multiple-point geostatistics

Ehsanollah Baninajar<sup>1</sup> · Yousef Sharghi<sup>1</sup> · Gregoire Mariethoz<sup>2</sup>

Published online: 29 October 2019  
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

## Abstract

Multiple-point statistics (MPS) have been widely used in Earth and environmental sciences because of their ability to generate realistic stochastic realizations of complex natural processes. The spatial patterns and statistical information required for MPS modeling are represented by a training image. However, each MPS algorithm has a specific set of parameters that have a direct impact on the quality of pattern reproduction and should be chosen prior to the modeling. While there are some general guidelines for some MPS algorithms, a general parameter interference methodology is currently lacking. To date, the common practice for finding optimal parameters is to carry out a sensitivity analysis, which can be cumbersome especially in complex applications. In this study, we propose the MPS Automatic Parameter Optimizer (MPS-APO), a generic method based on stochastic optimization to rapidly approximate optimal parameters for any MPS method and different types of settings. The MPS-APO formulates an objective function that quantifies spatial pattern reproduction for each set of parameters. The Simultaneous Perturbation Stochastic Approximation (SPSA) optimization method is used because of its computational efficiency, and also its ability to cope with the stochastic nature of the objective function. The optimization proceeds in two steps. The first step aims to optimize the parameters for the best quality regardless of computational cost. When no more improvement can be achieved, the second step minimizes the CPU cost without degrading the spatial structures reproduction attained in the first step. In this study, MPS-APO is performed on different pixel-based and patch-based MPS methods: SNESIM, FILTERSIM, Direct Sampling and Image Quilting. Test cases show that MPS-APO is a useful heuristic to automatically approximate optimal parameters for good patterns reproduction with minimal computational cost. Therefore, it can help non-expert users and increase the usability of MPS methods for practical applications.

**Keywords** Geostatistics · Multiple-point statistics · Parameterization · Parameters · Optimization · SPSA

## 1 Introduction

Multiple-point statistics (MPS) methods have attracted considerable attention in recent years. MPS is capable of reproducing complex patterns by combining the strength of pixel-based methods in data conditioning while also having an ability to generate large-scale structures comparable to object-based approaches. While the traditional geostatistical methods are limited to second order statistics like the covariance, multiple-point statistics approaches use higher order statistics, which are obtained from a training image (TI). The TIs are generally conceptual models from which spatial continuity, pattern and features are borrowed for simulations. In general they should be large enough to ensure that all patterns are sufficiently represented for

---

✉ Yousef Sharghi  
sharghi@sut.ac.ir

Ehsanollah Baninajar  
e\_baninajar@sut.ac.ir

Gregoire Mariethoz  
gregoire.mariethoz@unil.ch

<sup>1</sup> Department of Mining Engineering, Sahand University of Technology, New Sahand Town, Iran

<sup>2</sup> Institute of Earth Surface Dynamics (IDYST), University of Lausanne, 1015 Lausanne, Switzerland

simulations, although approaches exist to enrich the TIs (e.g., Rezaee et al. (2015); Straubhaar et al. (2019)).

Guardiano and Srivastava (1993) laid down the foundation of Multiple-point statistics decades ago, however the proposed algorithm was impractical due to its very high computational cost. Since then, further algorithms have been developed to improve upon these shortcomings. Multiple-point statistics techniques available in the literature include among others SNESIM (Strebelle 2002), FILTERSIM (Zhang et al. 2006), SIMPAT (Arpat and Caers 2007), DISPAT (Honarkhah and Caers 2010), Direct Sampling (Mariethoz et al. 2010), IMPALA (Straubhaar et al. 2011), HOSIM (Mustapha and Dimitrakopoulos 2011), CCSIM (Tahmasebi et al. 2012), Image Quilting (Mahmud et al. 2014), and optimization-based approaches (Kalantari and Abdollahifard 2016; Pourfard et al. 2017; Yang et al. 2016). These MPS algorithms use different approaches to read the TI, store information and reproduce the patterns. They can be divided into pixel-based and patch-based approaches, both having their own advantage and shortcomings. A qualitative comparison of these MPS methods can be found in Mariethoz and Caers (2014). MPS has been used in many applications such as reconstruction of porous media (Zhang et al. 2015), simulation of a petroleum reservoir (de Carvalho et al. 2017), time series modeling (Oriani et al. 2018), or for enhancing the spectral resolution of satellite images (Gravey et al. 2019).

Similar to other geostatistical methods, MPS require a statistical prior model to generate realizations. In two-point simulations, these statistics are taken into account by a mathematical formulation (e.g., variogram; Chiles and Delfiner (1999)). With MPS methods, the relevant statistics come from the TI, but also depend on some user-dependent choices of algorithmic parameters. Therefore, finding appropriate tuning parameters for each MPS method is a crucial issue to guarantee reasonable results. Failing to do so may directly affect the quality of pattern reproduction as well as CPU cost. Thus far, the selection of an optimal set of parameters associated with a TI has been done by extensive sensitivity analysis for each parameter, or by cumbersome trial and error tests followed by inspection of the ability to reproduce the patterns and large scale structures in the realizations. This process requires a comprehensive understanding of the algorithm and can be daunting for non-expert users. The fact that each MPS method requires a specific parameterization expertise also makes it difficult to compare different algorithms quantitatively, because it is hard to find the best set of parameters for each compared approach.

In order to deal with this problem, extensive sensitivity analysis have been done and sets of empirical guidelines have been proposed for parameterization of some MPS methods. For example Liu (2006) provides a complete

guide to SNESIM parameters and the effect of each parameter on pattern reproduction quality and CPU cost. A similar work was also done by Meerschman et al. (2013) for Direct Sampling (DS). However, for obtaining certain parameters like the weight of variables in multivariate cases (Jha et al. 2013), weights between soft and hard data (Rezaee and Marcotte 2017) or common MPS parameters in each application (e.g., Rezaee et al. (2014)), there is a lack of commonly accepted procedures to obtain a suitable set of parameters besides trial and error.

In recent years, some efforts have been made to find optimal parameters and reduce user dependency by selecting the parameters automatically. Template size is a very important parameter in patch-based methods. Honarkhah and Caers (2010) proposed DISPAT which is less sensitive to user-provided parameters. In this method, an optimal template size is automatically selected by calculating the mean entropy of the patterns for different template sizes and then using the inflection point of a so-called elbow plot. Strebelle and Cavelius (2014) proposed a method to minimize the computational cost of SNESIM while maintaining the pattern reproduction quality by optimizing the size of data templates. In this approach, the minimum acceptable data template size is chosen by a threshold beyond which additional conditioning data do not improve the estimated conditional probabilities. Kolbjørnsen et al. (2014) developed an automatic method to quantitatively determine the optimal number of multiple grids to be used in Markov mesh modeling which uses directional correlation functions obtained from the TI. However, the initial grid level should be chosen manually because the method cannot robustly estimate it for arbitrary TIs. Bai et al. (2016) also proposed a quantitative method to estimate a minimal number of multiple grids for the SNESIM algorithm. It uses the degree of spatial association quantified by modified joint count statistics in order to find the coarsest scale and consequently estimates the minimal number of multiple grid based on the criterion that the simulation scale should be no less than the target scale. Most recently, Dagasan et al. (2018) proposed an approach to automatically tune the parameters of DS. In this approach, the dissimilarity between the pattern histograms in conditional data and simulated image is quantified by calculating their divergence as an objective function. Then the optimal parameters are obtained by minimizing the objective function using simulated annealing. Dagasan et al. (2019) used this automatic tuning approach in an application to simulate ore boundaries in a lateritic bauxite deposit.

In similar simulation-based problems, the common way of obtaining optimal parameters is using optimization methods for the simulation analysis. Parameter optimization is the process of determining a set of parameter values

which are optimal in some desired sense (e.g., minimize/maximize the objective function under given constraints and over a given dataset). The criterion used to select the optimal parameters in this study is based on the pattern reproduction in the simulated image and the CPU cost of the simulation. Thus far, several metrics have been proposed to assess multiple-point pattern reproduction characteristics (Boisvert et al. 2010; Renard and Allard 2013; Tan et al. 2014). However, formulating an objective function for MPS is difficult because many of the criteria used to assess pattern reproduction are based on the visual inspection of the results. Another important limitation is the computational cost of MPS, therefore the selected optimization method should converge on a solution with acceptable amount of resources. To make matters worse, geostatistical simulations are stochastic, which can make the optimization converge to a false solution or not converge at all.

In this study we introduce the MPS Automatic Parameter Optimizer (MPS-APO), a generic method based on stochastic optimization to rapidly determine acceptable parameters, in different settings and for any MPS method. The method relies on a cross-validation based on the TI to quantify pattern reproduction. The proposed algorithm formulates an objective function by placing random gaps in the TI and simulating these gaps using the chosen MPS method. This approach is very similar to the widely used cross-validation in classical geostatistics. In situations like spatiotemporal reconstruction of gaps in multivariate fields (e.g., satellite images), the known domains have been already used as TI to reconstruct the missing parts (Mariethoz et al. 2012).

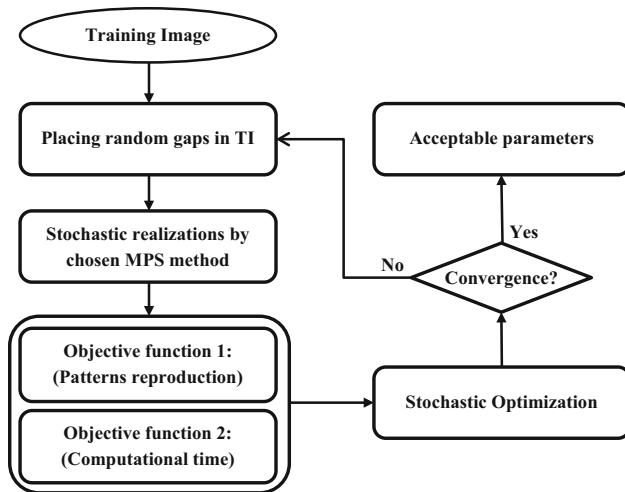
To overcome convergence issues, the Simultaneous Perturbation Stochastic Approximation (SPSA) method is used to deal with the stochastic nature of the objective function and for its computational efficiency. SPSA, first introduced in Spall (1987), is a stochastic optimization algorithm useful in difficult multivariate optimization problems where the objective function calculation is costly or corrupted by noise. It has been widely used in statistical parameter estimation (Kothandaraman and Rotea 2005), simulation-based optimization (Gosavi et al. 2007), pattern recognition (Gu et al. 2012), signal processing (Zonoobi et al. 2011), or neural network training (Abdulsadda and Iqbal 2011). With an objective function formulated as the mean square errors between the simulated and original gaps values, SPSA optimization proceeds in a 2-step approach. First the optimization converges to a set of parameters with a best reproduction of patterns regardless of CPU cost. Then, when no significant improvement can be brought in the reproduction quality, the second optimization step starts, which minimizes the CPU cost without degrading the reproduction quality attained in the first step.

The rest of this paper is organized as follows. First, we describe the objective function that is used for quantification of simulations quality and also computational cost. Next, the rationale behind choosing a stochastic low-cost optimization method to obtain an satisfying solution is discussed in Sect. 2.2. Then, MPS-APO's performance is demonstrated in Sect. 3 by performing it on four different geostatistical methods: SNESIM, FILTERSIM, DS and Image Quilting (IQ) to show the generality of the approach. For each method, a different test case is used to automatically determine optimal parameters. A standard test case is then used to compare these methods with automatically determined optimal parameters. The paper is finally summarized in Sect. 4. Source code and documentation for the MPS-APO is available at: <https://github.com/EhsanBani/najar/MPS-APO>.

## 2 Methodology

In geostatistical simulations, input parameters are typically inferred prior to any model generation. In classical geostatistics, cross-validation is widely used to justify these choices. With this useful tool, the impact of each set of input choices (i.e., variogram models, weighting procedures, search strategies, etc.) can be assessed and quantified. Through cross-validation, models can be evaluated and optimized. The cross-validation process consists of temporary removal of a known data point from the dataset and using the surrounding samples to estimate it using a given estimation method. This approach allows the comparison of true and estimated values by using available information in the dataset. However, there is currently no equivalent of cross-validation for MPS.

The proposed MPS-APO algorithm compares multiple-point statistics between realizations made by MPS algorithms and its associated TIs, and tries to minimize the differences by optimizing the parameters of these algorithms. Because the target multiple-point statistics are explicitly contained in the TI, the pattern reproduction can be quantified by calculating the mismatch between the simulated gap values and the original TI values. The objective function is formulated such as to minimize this reproduction error. The main assumption is that a parameterization that can correctly recover the missing gap values is able to adequately capture the TI characteristics. The implementation of the optimization algorithm is divided in a number of steps which are described in detail in the following sections. Figure 1 represents the flowchart of the MPS-APO Algorithm.



**Fig. 1** The flowchart of MPS-APO algorithm

## 2.1 Quantification of patterns reproduction

Our methodology relies on this definition: for each MPS algorithm, a specific set of input parameters is better than others if it allows reproducing the input data statistics better and at a lower computational cost. In order to quantify multiple-point simulation performance, we randomly place a number of gaps in the TI with only two constraints:

1. The gaps should not have any overlaps and preferentially placed far from each other.
2. Gaps should be preferentially placed in locations of highest variance because these are often area that are critical to reproduce correctly.

In order to avoid using known information in simulations, the values in the gap locations are removed from the TI and are replaced by null values. This new TI altered with gaps is then used as an input to simulate the gaps. For each simulation, the mismatch between the original pixels in the TI and the simulated ones is calculated for each gap location. The fact that gap values are removed from the training data means that the simulation can never exactly match the original data, making it impossible for the phenomenon of verbatim copy to occur. This is further discussed in Sect. 2.4. These gaps can have any shape, however the shape should provide sufficient conditioning so that the original values can be reproduced exactly. For example, very large gaps will result in unconditional simulation of large areas that do not allow assessing the goodness of a specific set of parameters. Overlaps in gaps may create such unintended large areas in the TI, and thus should be avoided. This is ensured by a user-defined parameter of minimum gap distance. In this study, simple square gaps (with 15 to 20 pixel length for each side) are

used for 2D TIs. For 3D cases, cubic gaps with 15 to 20 pixel sides are used. The number of random gaps depends on the size of input TI and its patterns shape and scale, but generally the amount of random gaps should be between 5 and 20% of TI size to avoid affecting the structures present in it. Large amount of gaps may lead to a loss of pattern information in the TI, and also affect computational cost because it represents a larger number of values to simulate. This gap conditioning strategy is used when optimizing the parameters of pixel-based algorithms. Figure 2 illustrates the gap approach for quantifying pattern quality reproduction for a continuous TI. The mismatch formulation is explained in details in Sect. 2.3.

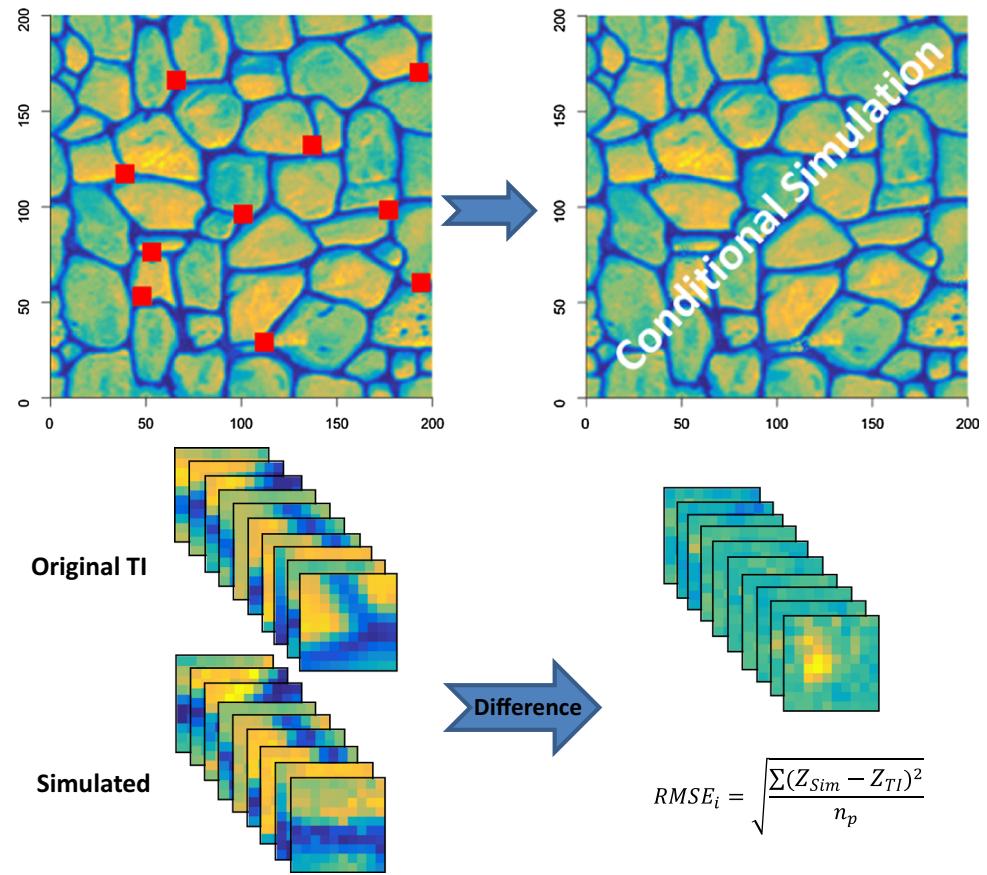
Conditioning using the entire TI (except for the removed gap values) has a computational advantage for pixel-based methods like DS, which are able to accommodate dense conditioning. This can considerably speed up the optimization process for these methods. However, this is not the case for patch-based methods, which generally have difficulties handling large amounts of conditioning data. For patch-based algorithms, a dispersed points strategy is adopted for conditioning, whereby instead of using the whole known dataset as a conditioning, in each simulation a smaller portion (e.g., 5%) of it is randomly selected to simulate the gap locations.

Any MPS method can be used to simulate the values removed from the gaps. Also, any error norm can be chosen to calculate the mismatch between original and simulated pixels. However, because geostatistical realizations are stochastic, each evaluation of the objective function will generate a slightly different outcome, resulting in a noisy objective function. Some steps are taken to deal with the stochastic nature of the objective function. First, to maintain consistency between simulation results, the gaps should have a similar configuration in each simulation. Therefore, the gap locations are chosen randomly among the locations with highest pattern variability in the TI. In the MPS-APO, the variability associated to each potential gap location is computed during initialization. It is defined as the standard deviation of the values within each gap:

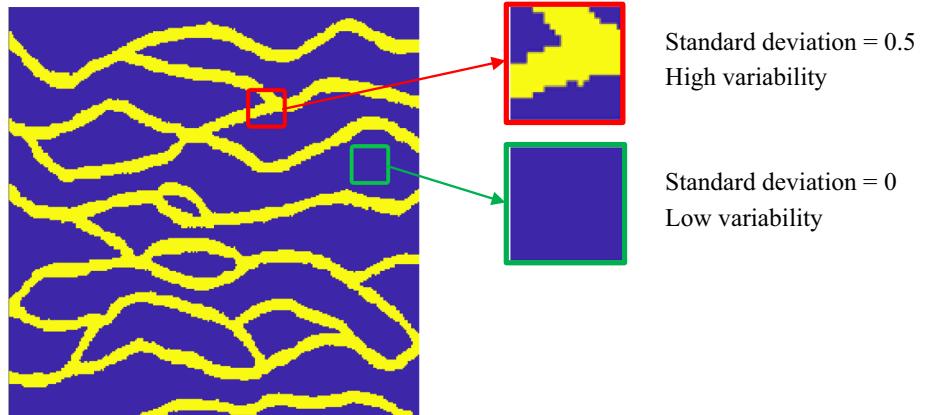
$$V = \sqrt{\frac{\sum_{i=1}^n (Z_i - \bar{Z})^2}{n_p - 1}} \quad (1)$$

where  $Z_i$  is the  $i$ th pixel of the gap,  $n_p$  is number of pixels in the gap and  $\bar{Z}$  is average of all gap pixels. For each simulation, only locations with a standard deviation within the two highest deciles are randomly selected and used for parameters optimization. Figure 3 illustrates the precomputed gap database and the associated pattern variability. Secondly, to reduce the objective function noise, an average of several realization with random gaps is used for each

**Fig. 2** Quantification of pattern quality reproduction for a continuous TI



**Fig. 3** Illustration of the gap variability that is precomputed for all possible gap locations



objective function evaluation. This process stabilizes the objective function, but comes with a higher computational cost. Moreover, the optimization should be able to handle noise in such measurements. Section 2.2 delves into these requirements and the reasons why SPSA optimization was chosen in this study.

## 2.2 Optimization

In the past few decades, many new optimization theories and algorithms have been proposed. Optimization methods

can be divided into deterministic and stochastic approaches. Deterministic optimization methods assume that all the information about the objective function (and for some methods its derivatives) are available, which are then used to determine the search direction at each iteration (e.g., Newton–Raphson method, Pattern Search (Hooke and Jeeves 1961), Nedler–Mead Simplex (Nelder and Mead 1965), NLPQL (Schittkowski 1986)). In cases where there is random noise in the measurements of the objective function, deterministic methods are ineffective because this noise can completely change the course of the optimization

with misleading information. In order to deal with this noise, different stochastic optimization algorithms have been developed (e.g., SPSA (Spall 1987), Simulated Annealing (Kirkpatrick et al. 1983), Particle Swarm Optimization (Eberhart and Kennedy 1995), Genetic Algorithms (Goldberg 1989)).

The simultaneous perturbation stochastic approximation (SPSA) performs gradient approximation with only two objective functions evaluations, regardless of the number of parameters to be optimized. This reduces the optimization cost significantly, especially when there are many of variables to be optimized.

Spall (1992) shows that with some conditions, SPSA almost surely converges to a local minimum with a sufficient number of iterations. Here the goal is to rapidly find one acceptable parameterization rather than a global minimum, therefore finding a local minimum can be acceptable.

The optimization problem can be formulated as finding an optimal solution ( $\theta^*$ ) such that

$$\theta^* = \arg \min_{\theta \in P} y(\theta) \quad (2)$$

where  $y(\cdot)$  is the objective function and  $\theta$  is a  $i$ -dimensional vector of input parameters for each MPS method. Each element of  $\theta$  is a number with an upper and lower bound defined by a parameter space  $P$  which is the set of possible values that the MPS method can take as input. The evaluations of the objective function should be available for every  $\theta$  in parameter space. Since the function has a stochastic nature, these measurements include noise. In stochastic gradient methods, no direct measurements of gradients are used. In order to approximate the gradient, at each iteration SPSA takes a random direction in the search space and measures the objective function in two evaluation points: one in the chosen direction and one in the opposite direction. In this way, an increase or decrease in the objective function can be determined, allowing estimation of the gradient:

$$\hat{g}_{k;i}(\hat{\theta}_k) = \frac{y(\hat{\theta}_k + c_k \Delta_{k;i}) - y(\hat{\theta}_k - c_k \Delta_{k;i})}{2c_k \Delta_{k;i}} \quad (3)$$

where  $\hat{g}_{k;i}(\hat{\theta}_k)$  is the  $i$ th component of the gradient approximation at the  $k$ th iteration,  $\hat{\theta}_k$  denotes the estimate for optimal  $\theta^*$  at the  $k$ th iteration with  $i$  elements,  $y(\cdot)$  denotes the objective function measurement,  $\Delta_{k;i}$  is the simultaneous perturbation vector which consists of  $i$  independent random variables generated by Monte Carlo and  $c_k$  denotes a sequence of positive scalars. A simple but effective optimal choice (Sadegh and Spall 1998) for  $\Delta_k$  components are to use symmetric Bernoulli distribution which is essentially a random switching between +1 and

−1 with a probability of 0.5 for each outcome. Also note that in (3), only two objective function measurements are needed for each iteration, regardless of the number of parameters, because the numerator does not change for each of the  $i$  components.

When  $\hat{g}_{k;i}(\hat{\theta}_k)$  is obtained, (4) is used to update the estimate of  $\theta$  to a new value of  $\hat{\theta}_{k+1}$ :

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}_k(\hat{\theta}_k) \quad (4)$$

where  $a_k$  is a sequence of positive numbers similar to  $c_k$ . The gain sequences ( $a_k$  and  $c_k$ ) control the step size and gradient in each iteration. They are normally used as a function of  $k$  and get smaller with  $k$  getting larger, in an annealing fashion. Therefore, their choice is very important and has a direct impact on the convergence of SPSA. The gain  $c_k$  can be set proportional to the standard deviation of the noise in the objective function. The gain of  $a_k$  together with the estimate of gradient  $\hat{g}(\hat{\theta}_k)$  controls the descent step size in each iteration. For example a matrix of different  $a_k$  can be used in cases where parameters have a very different magnitude. One reason for selecting SPSA for optimization (beside its efficiency and speed) is its straightforward parameterization, therefore avoiding hyperparameters. Extensive studies have been conducted on their selection and also the respective algorithm coefficients. At a given iteration  $k$ ,

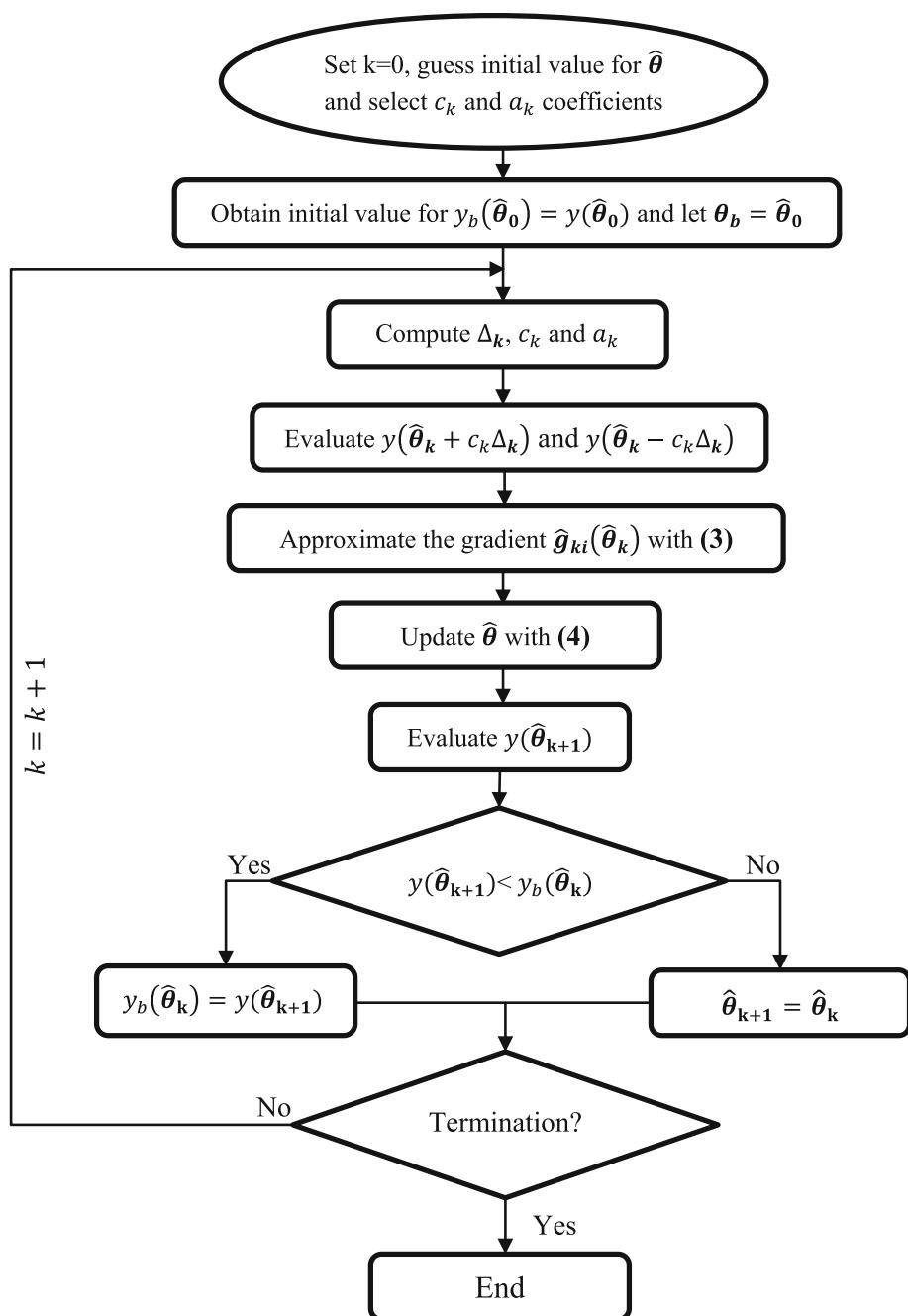
$$a_k = \frac{a}{(A + k + 1)^\alpha} \quad (5)$$

$$c_k = \frac{c}{(k + 1)^\gamma} \quad (6)$$

where  $\alpha = 0.602$ ,  $\gamma = 0.101$ ,  $c$  is the standard deviation of the measurement noise,  $A$  is chosen as a value  $\leq 10\%$  of the maximum number of iterations and  $a$  is the smallest initial change desired in parameters. In-depth guidelines regarding the choice of gain sequences can be found in Spall (1998).

In the implementation of SPSA used in this study, an adaptation called “blocking” is also employed to speed up convergence and enhance the stability of the optimization. This adaptation measures an additional objective function at  $\hat{\theta}_k$  in addition to the two perturbations, and blocks the updating of  $\hat{\theta}_k$  if the new objective function value is significantly worse than the current one. However, this extra function evaluation increases the optimization cost by 33% (Spall 1998). Figure 4 represents the flowchart of the SPSA algorithm used in this study.

**Fig. 4** SPSA algorithm flowchart



### 2.3 Formulation as an optimization problem

MPS-APO proceeds in two steps. In the first optimization step, parameters with the best patterns reproduction are obtained, regardless the CPU cost. The second optimization step starts when no significant improvement can be brought in the pattern reproduction. This second step aims at minimizing the CPU cost of the simulation without degrading the reproduction quality attained in the first step.

- Step 1: Quality reproduction

An initial guess should be provided for parameters as a starting point, then the optimization iteratively improves the initial guess in an attempt to converge to an optimal solution. Since the gaps are relatively small, they are affected by the conditioning data surrounding them, and therefore an optimal parameterization should in average provide the best possible reproduction of the gaps data. The Root Mean Square Error (RMSE) is used to quantify these prediction errors. RMSE is a frequently used measure of the differences between predicted values and the values actually observed. The RMSE for each gap is calculated as

$$RMSE_i = \sqrt{\frac{\sum(Z_{Sim} - Z_{TI})^2}{n_p}} \quad (7)$$

where  $i$  goes from 1 to number of gaps in each simulation,  $Z_{Sim} - Z_{TI}$  is the difference between original and simulated pixel value and  $n_p$  is the number of pixels in each gap. For continuous TIs,  $Z_{Sim} - Z_{TI}$  can be easily calculated by subtracting the simulated and original values for each pixel (Fig. 2). For categorical TIs we use the Hamming distance, whereby if the category is simulated correctly then  $Z_{Sim} - Z_{TI} = 0$  otherwise  $Z_{Sim} - Z_{TI} = 1$ .  $RMSE_T$  is the average RMSE of all gaps, which is used in MPS-APO as our objective function (8).

$$RMSE_T = \frac{\sum RMSE_i}{n_g} \quad (8)$$

where  $RMSE_i$  is RMSE of the  $i$ th gap and  $n_g$  is the number of gaps in each simulation. At each optimization iteration,  $RMSE_T$  is calculated and the parameters are updated accordingly. This process is then repeated for the updated parameter values until a stopping criterion is met. We use relatively simple stopping criteria: the optimization is terminated when the improvement in objective function is less than 0.05 for 5 successive iterations or if the maximum number of iterations is reached (e.g., 50 iterations). The final parameters values represent an approximation of the optimal parameters for best quality reproduction.

In multivariate cases, different types of TIs are used as input (i.e., categorical, continuous) and the weight of each variable has an effect on the final simulation quality. For these cases, the objective function is the sum of the RMSE for all variables. To this end, the RMSE for each TI should be in the same range and comparable. In order to obtain weights that can be compared between variables, each RMSE is normalized by its interquartile range (9).

$$RMSE_T = \sum_{i=1}^{n_v} \frac{RMSE_i}{IQR(TI_i)} \quad (9)$$

where  $RMSE_i$  is the total RMSE for the  $i$ th variable,  $TI_i$  is the  $i$ th variable of the TI and  $n_v$  is the number of variables in the TI.

- Step 2: Minimization of computational cost

In this step, the objective function becomes the CPU time taken for performing each function evaluation. However, in many cases the parameters that minimize CPU time also degrade the simulation quality. Therefore, in order to avoid significant quality degradation, a term is used to keep the RMSE under a user defined threshold (i.e., 10% of best quality RMSE found in step 1). This threshold specifies the tradeoff between reproduction quality and computational cost. If the RMSE goes beyond this

threshold, the parameters are not updated for that iteration. With the same stopping criterion as in step 1, the optimization stops if the maximum number of iteration is reached, or when no improvement can be brought to the CPU time without degrading the quality too much for five successive iterations. The final parameters values represent an approximation of the optimal parameters that combine good TI patterns reproduction and for minimal CPU cost.

## 2.4 Dealing with verbatim copy and artifacts

In MPS simulations, the exact copying of the training image into the simulation grid (known as verbatim copy) is generally not desired because it artificially reduces the variability between realizations, and therefore is contrary to the aim of geostatistics (i.e., modeling the spatial uncertainty; Mariethoz and Caers (2014)). Several steps have been taken in MPS-APO to prevent parameterizations that result in verbatim copy. For every simulation, the gap values data are taken out of the dataset and replaced by null values. It is this new dataset that is used as TI, making it by construction impossible for verbatim copy to occur. This approach is used for MPS implementations that allow using incomplete TIs. In methods where this is not possible, preventing verbatim copy is achieved by selecting parameter bounds such that there is always a possibility to create new patterns. For example with IQ a very small number of replicates (typically 1) can result in verbatim copy (Mahmud et al. 2014). Such settings are therefore prohibited. The same is true for tree-based or list-based methods (e.g., SNESIM and FILTERSIM).

We also expect artifacts to occur near the conditioning data, which is a known phenomenon inherent to some MPS approaches (Gardet et al. 2016; Saripally and Caers 2008). Another reason often mentioned for the occurrence of artifacts in MPS is significant differences between the spatial statistics of local conditioning data and the TI (Boisvert et al. 2007). In MPS-APO, the TI is representative of the conditioning data because the conditional data come from it, preventing such incompatibilities. In any case, as we calculate the difference between the simulation and the training Image, any artifacts resulting in larger errors should affect the objective function, and therefore be minimized during optimization.

## 3 Results and discussion

To demonstrate the applicability of MPS-APO, different examples are presented in this section using SNESIM, FILTERSIM, DS and IQ. In Sects. 3.3–3.6, MPS-APO is performed on different types of TIs to show the generality of the proposed approach. In Sect. 3.9, a same TI is used

for all MPS methods to allow a fair comparison between MPS simulation methods. While any implementation on MPS can be used with MPS-APO, in this study we use SGeMS (Remy et al. 2009) for SNESIM and FILTERSIM, an in-house implementation of Direct Sampling (Mariethoz et al. 2010) and the Image Quilting code of Mahmud et al. (2014) available at <https://github.com/GAIA-UNIL/Image-Quilting>.

### 3.1 Parameters to be optimized

Generally any parameter that affects the simulation quality and computation time can be used in the optimization. The optimized parameters, their bounds and initial values are provided in Table 1. In our tests, only a limited number of parameters that have direct effects on TI reproduction are selected for optimization. For the other parameters (e.g., target proportion and servosystem correction or rotation and affinity ratios), the algorithm's default value is used. However if required, such parameters could also be included.

### 3.2 The training images

Figure 5 shows the TIs used for testing MPS-APO. TIs used in this study are categorical, continuous, multivariate and 3D to show that the MPS-APO can work with all types of TIs. Figure 5a is a marble thin section (Meerschman et al. 2013) segmented in 3 categories, with a size of  $408 \times 335$  pixels. Figure 5b is Stanford V dataset with a size of  $60 \times 50 \times 40$ . The Stanford V dataset (Lee and Mukerji 2012) is a synthetic continuous 3D model set in a

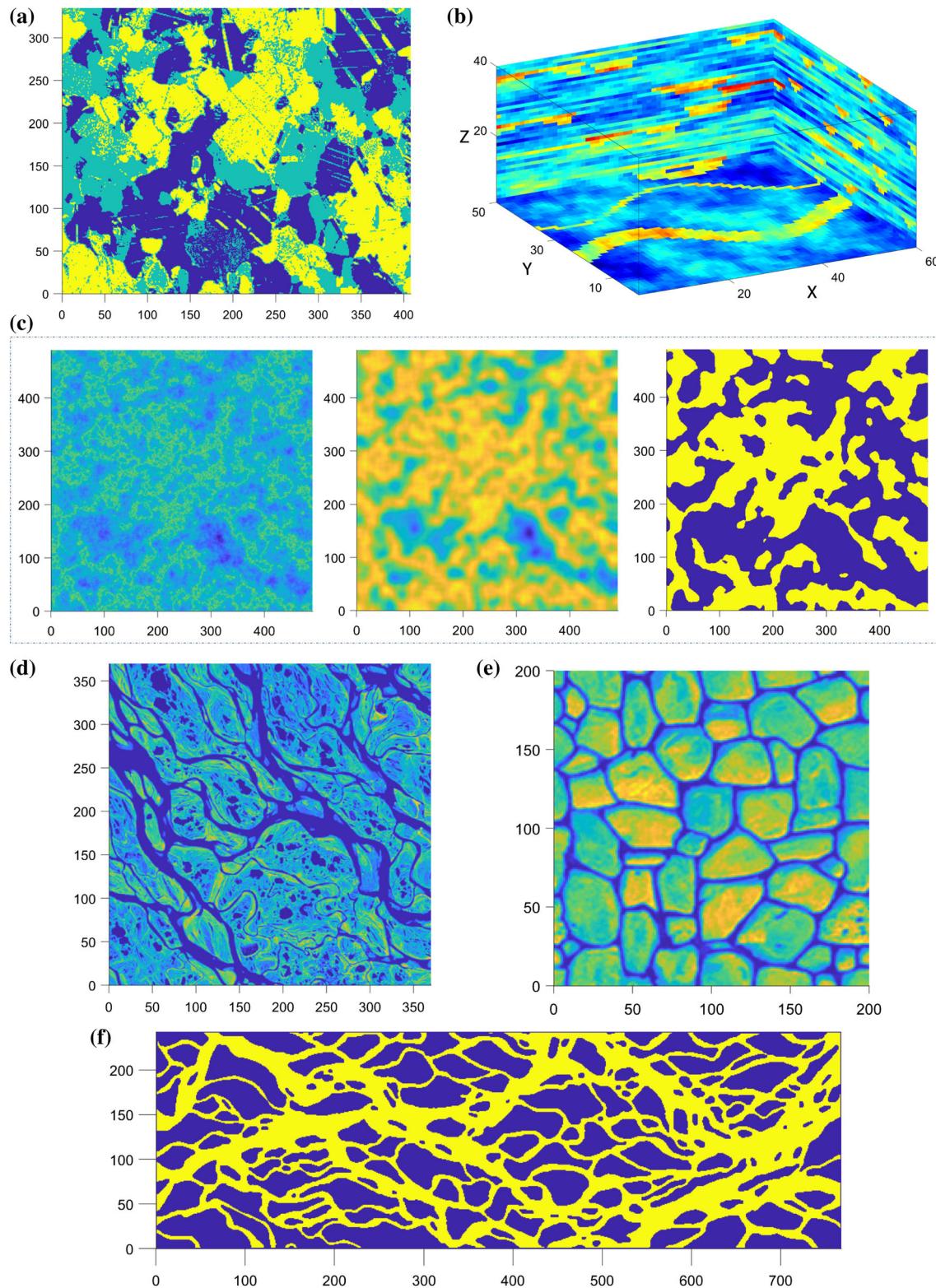
fluvial environment, containing porous sand channels surrounded by shale, and representing a clastic reservoir. Figure 5c is a synthetic 3-variate TI (two continuous and one binary) with the size of  $490 \times 490$  pixels. Figure 5d is a Landsat satellite image of the Lena River delta (Russian federation) representing a complex fluvial system with  $370 \times 370$  pixels. The Stonewall (Zhang et al. 2006) in Fig. 5e is a continuous TI with a size of  $200 \times 200$  pixels which represents packed stones. Figure 5f is a  $768 \times 243$  binary TI based on the satellite image of the Ganges delta in Bangladesh (Mariethoz and Caers 2014). It represents a braided river classified into channels (yellow) and alluvial bars (dark blue). In the following, each MPS method has its parameters optimized with a different TI. In addition and to a uniform comparison, parameters optimization is carried out for all methods using the TI of Fig. 5f.

### 3.3 SNESIM

MPS-APO is applied to SNESIM and performed on the marble thin section shown in Fig. 5a and Bangladesh TI shown in Fig. 5f. In the marble thin section example, 15 gaps of size  $20 \times 20$  pixels are used for each simulation. For each objective function evaluation, the average of 10 realizations with random gaps are generated to average out the stochastic fluctuations. For the Bangladesh example, 50 gaps with the size of  $20 \times 20$  pixels are used for each simulation. For each objective function evaluation, the average of 5 realizations is used for each simulation. Figure 6a shows the boxplots of the RMSE in each iteration, which gives an idea about the quality reproduction in each iteration. The boxplot of computational cost for each

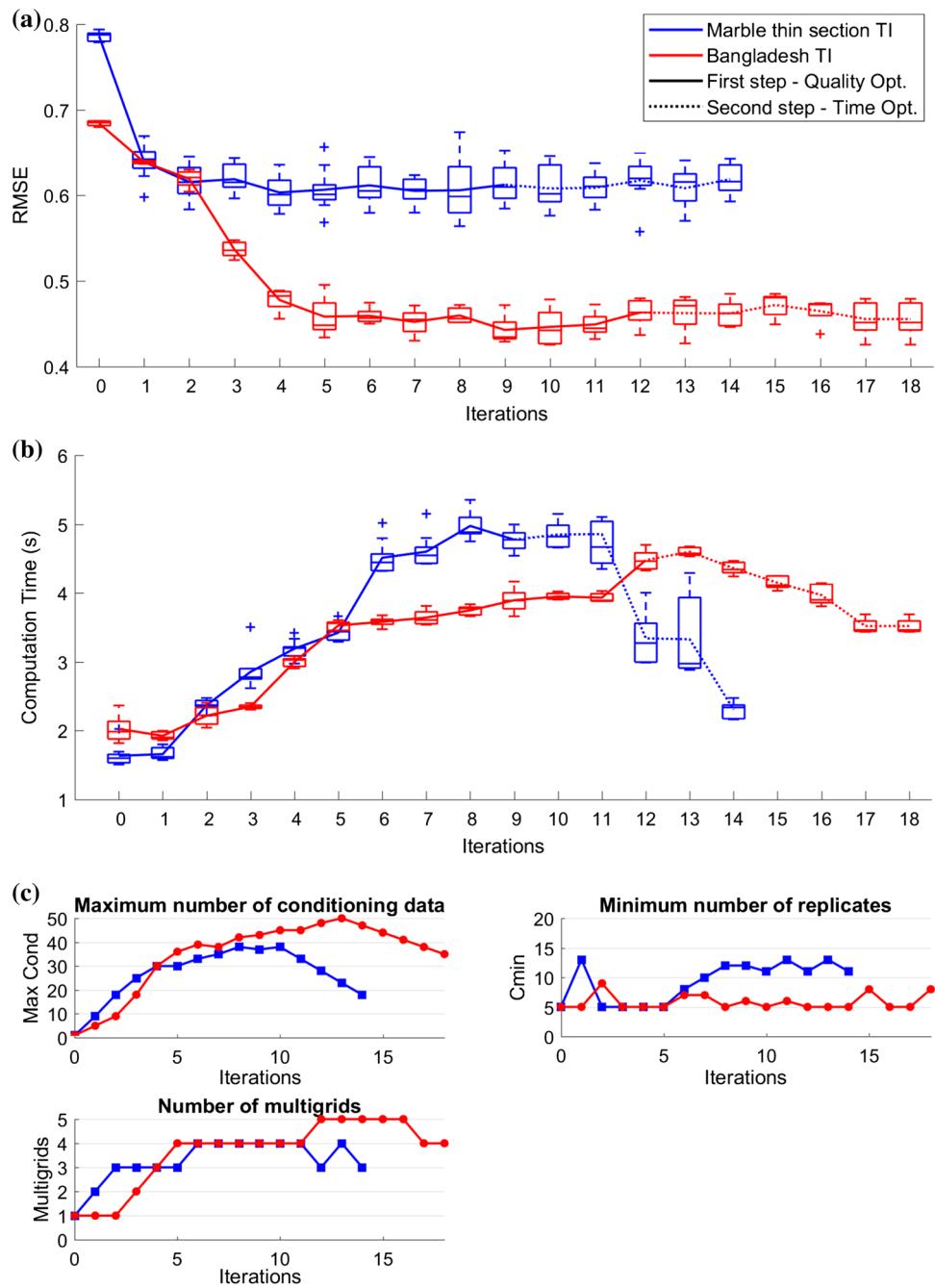
**Table 1** Parameters optimized for each MPS method

Methods	Parameters	Initial value	Bounds
SNESIM	Maximum number of conditioning data	1	[1 100]
	Minimum number of replicates	5	[5 60]
	Number of multiple grids	1	[1 10]
FILTERSIM	Search template	1	[1 30]
	Inner patch	1	[1 30]
	Number of replicates	5	[5 100]
Direct Sampling	Number of multiple grids	1	[1 3]
	The distance threshold	0.3	[0.3 0.001]
	The fraction of scanned TI	0.1	[0.1 1]
Image Quilting	Maximum number of closest neighbors	1	[1 100]
	Relative weight of each variable (for multivariate cases)	0.33, 0.33, 0.34	[0 1]
	Patch size	20	[10 100]
	Overlap size	4	[4 70]
	Number of replicates	10	[3 20]



**Fig. 5** Training images that are used in this paper: **a** categorical TI of a marble thin Section (3 categories), **b** continuous TI of 3D channels, **c** multivariate TI with two continuous and one binary variables, **d** continuous TI of Lena delta, **e** continuous TI of Stonewall and **f** binary TI of the Ganges Delta (Bangladesh)

**Fig. 6** MPS-APO performed on SNESIM with Marble thin section TI (Fig. 5a) and Bangladesh TI (Fig. 5f). **a** Quality reproduction, **b** computational cost and **c** parameters space at each iteration



iteration is also shown in Fig. 6b. After the optimization progresses and decreases to a point where no further improvement can be brought to RMSE (Fig. 6a), the second step kicks into minimize the computational time. The

second step is shown with a dotted line. To avoid degrading the quality while optimizing the time, a 10% threshold of the best RMSE is defined and used in all the examples in this study. Figure 6c illustrates how the optimization

**Table 2** Optimal parameters for quality and time for SNESIM and TIs shown in Fig. 5a and f

SNESIM parameters	Marble thin section TI		Bangladesh TI	
	Quality	Time	Quality	Time
Maximum number of conditioning data	37	18	45	35
Minimum number of replicates	12	11	6	8
Number of multiple grids	4	3	4	4

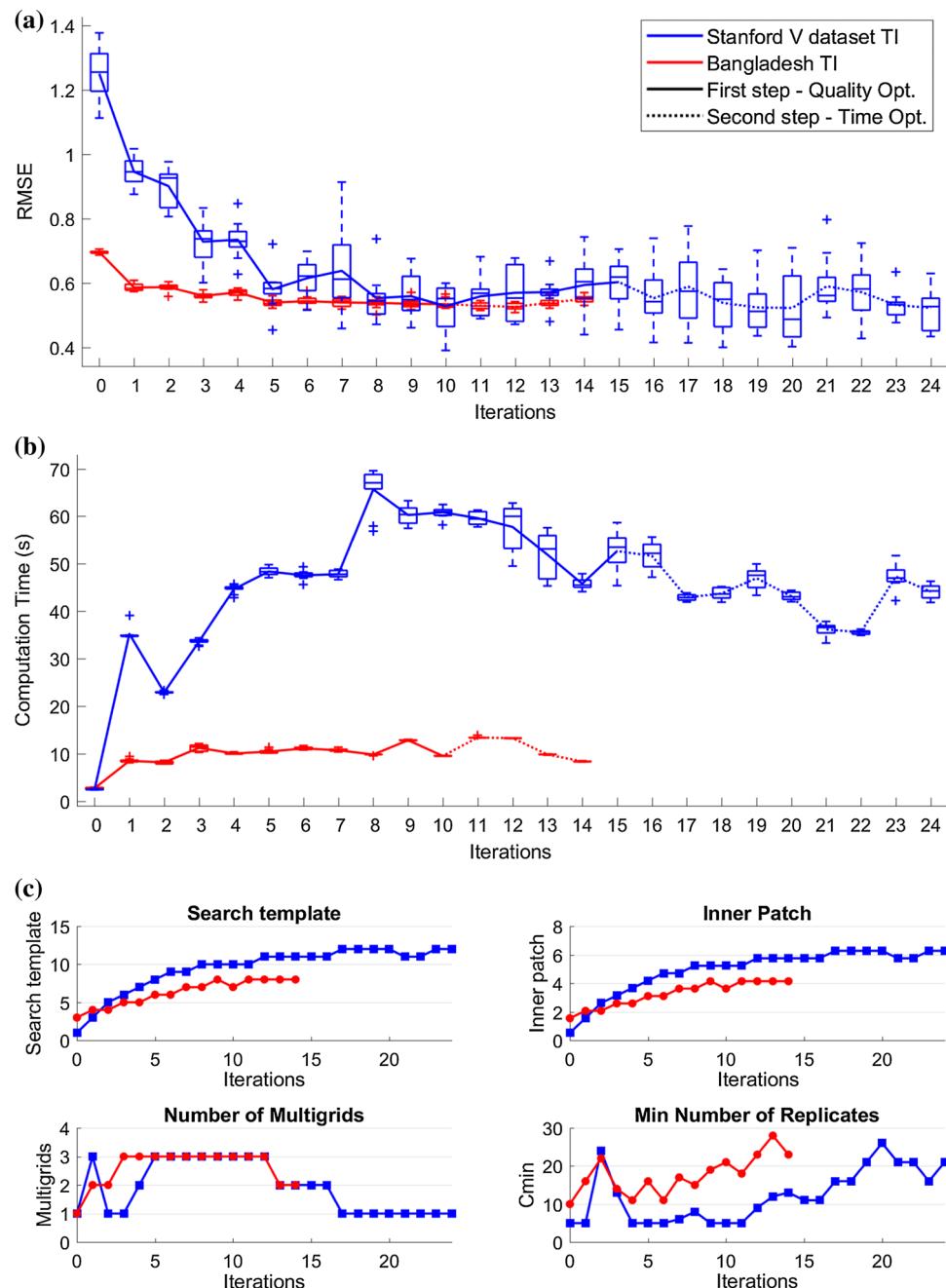
searches the parameters space and Table 2 shows the resulting approximation of optimal parameters at the end of each step. In step 2, the computational time is reduced by a factor of 2 without making a significant difference in the RMSE. The resulting parameters are in agreement with the guidelines in Liu (2006).

### 3.4 FILTERSIM

The automatic parameterization of FILTERSIM is performed using the Stanford V dataset (Fig. 5b) and

Bangladesh TI (Fig. 5f). In case of Stanford V dataset, due to the small size of the TI for each simulation only 3 cubic gaps of size  $15 \times 15 \times 15$  are used. However for Bangladesh TI, 50 gaps with the size of  $20 \times 20$  pixels are used for each simulation. In both examples, the average of 10 realizations are computed for each objective function evaluation. Figure 7 demonstrates the relationship between parameters, quality and time. Table 3 shows the approximation of the optimal parameters at the end of each step.

**Fig. 7** MPS-APO performed on FILTERSIM with Stanford V dataset TI (Fig. 5b) and Bangladesh TI (Fig. 5f). **a** Quality reproduction, **b** computational cost and **c** parameters space at each iteration



**Table 3** Optimal parameters for quality and time for FILTERSIM and TIs shown in Fig. 5b and f

FILTERSIM parameters	Stanford V dataset TI		Bangladesh TI	
	Quality	Time	Quality	Time
Search template	$11 \times 11 \times 9$	$13 \times 13 \times 9$	$7 \times 7 \times 5$	$9 \times 9 \times 7$
Inner patch	$9 \times 9 \times 7$	$9 \times 9 \times 7$	$3 \times 3 \times 1$	$5 \times 5 \times 3$
Number of replicates	11	16	21	23
Number of multiple grids	2	1	3	2

### 3.5 Direct Sampling (DS)

The DS algorithm is able to account for multiple-dependencies between several variables, therefore these variables can be jointly simulated or be used as conditioning data. However in multivariate cases, in addition to general DS parameters, the user defines a weight for each variable. Automatic parameterization for DS is performed on a the 3-variate TI of Fig. 5c and Bangladesh TI of Fig. 5f. In the multivariate example, for each simulation 15 gaps of size  $15 \times 15$  pixels are randomly placed in the TI. In multivariate cases, the objective function could be formulated such as to minimize the average error of all variables, however in this example the objective function is set to minimize the error only for the first variable. For the Bangladesh TI, 50 gaps of size of  $20 \times 20$  pixels are used for each simulation. In both examples, the average of 5 realizations are computed for each objective function evaluation. The relationship between parameters, quality and time are demonstrated in Fig. 8 and approximation of optimal parameters for each step are shown in Table 4. The distance threshold and the number of closest neighbors are the most important parameters for DS. However, Fig. 8 also shows that choosing large values for these parameters does not always result in better quality. For example, in case of multivariate TI, the computational time for the final parameters is improved significantly in comparison to the step 1 parameters. This shows the importance of considering both patterns reproduction and computational cost.

### 3.6 Image Quilting (IQ)

Automatic parameterization for IQ is performed on a Landsat satellite image of the Lena River delta (Fig. 5d) and Bangladesh TI (Fig. 5f). To speed up MPS-APO, each objective function evaluation is based on only one realization. This causes fluctuations in the objective function (Fig. 9a). However, because of SPSA's efficiency in handling noisy measurements, the optimization is still able to converge. In IQ or any patch-based method, conditioning is always a challenge and sometimes dense conditioning is computationally cumbersome, not necessarily leading to better results. In the Lena example, 1000 random points

taken from the TI are used for conditioning instead of the dense conditioning around the gaps.

As Fig. 9 suggests, larger patch size leads to better reproduction of large-scale features and subsequently lower RMSE for faster simulation time. However, using very large patches will result in difficult conditioning on denser datasets and also occurrence of verbatim copy. Mahmud et al. (2014) suggests that a high number of replicates ( $> 5$ ) and a larger overlap area leads to non-unique patch selection and new patterns along the minimum error boundary and therefore results in less verbatim copy. Table 5 shows the approximation of optimal parameters for the Lena TI and Bangladesh TI with different conditional dispersed points.

In order to determine the sensitivity of patch-based methods like IQ to the number of dispersed points used in conditioning, a test is performed with 1000 and 5000 dispersed points on the  $768 \times 243$  binary TI of the Ganges delta (Fig. 5f). Figure 9 show the optimization process and the final parameters for each case. Figure 9a suggests that both cases have similar RMSE. While both have the same quality, the number of conditioning data has a direct effect on the computational time (Fig. 9b).

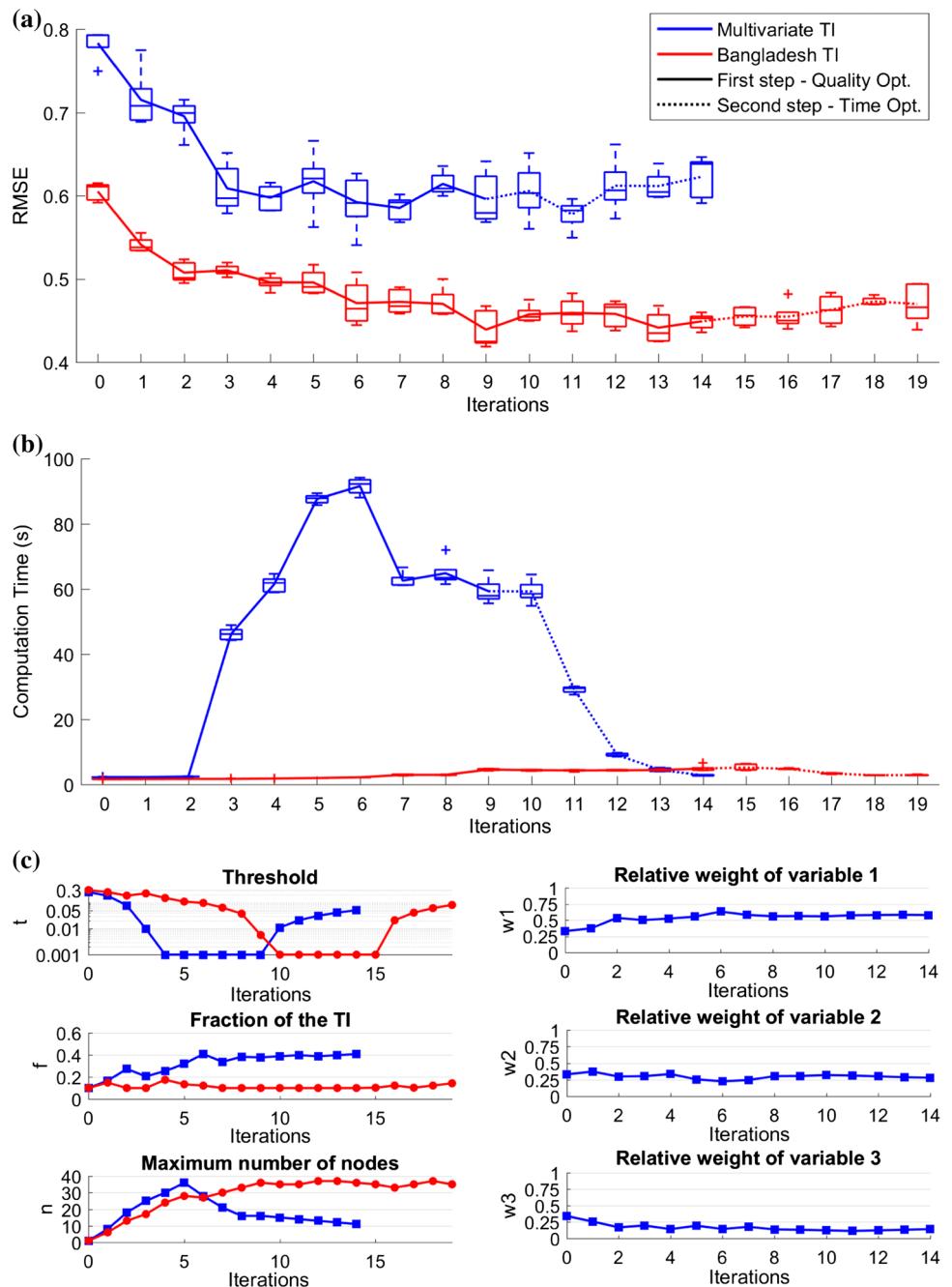
### 3.7 CPU time for MPS-APO

The optimization method used in this paper approximates an optimal solution using an iterative process because SPSA only evaluates three objective functions per iteration, which helps reduce the total CPU cost. Table 6 shows the total cost of MPS-APO for each simulation method in the examples shown in Sects. 3.3–3.6.

### 3.8 Sensitivity to gap size and quantity

A sensitivity analysis of gaps size and quantity is performed to assess their importance. Figure 10 shows the gap settings that are used in the optimization. Square gaps with three different sizes and three different densities for each size are used in the optimization. All simulations are performed on a  $768 \times 243$  binary TI of the Ganges delta (Fig. 5f). The SNESIM algorithm is used to simulate the gaps. Gaps of Fig. 10 are randomly placed on the dataset for each simulation.

**Fig. 8** MPS-APO performed on direct sampling with multivariate TI (Fig. 5c) and Bangladesh TI (Fig. 5f). **a** Quality reproduction, **b** computational cost and **c** parameters space at each iteration



If the gap sizes and densities have an effect on the optimization process, the results of the parameterization should change for each case. The optimizations for all cases are illustrated together in Fig. 11. The Fig. 11c clearly shows that the parameters are similar for all cases, therefore the gaps size and density do not have a large influence. Our tests shows that the gap size only affects the optimized parameters when the proportion of uninformed nodes is much smaller than 5%. An option is provided in the MPS-APO code for the users to use the proportion of

the unknown data to the TI instead of gap size and gap number.

Figure 11a also shows a direct relationship between the gap size and the RMSE. The  $10 \times 10$  gaps, regardless of gap density, have the lowest RMSE in Fig. 11a. Larger gap sizes have noticeably higher RMSE. This is important because it shows that larger gaps may contribute to the already noisy objective function and prevent the optimization from converging. While small gaps have benefits (e.g. working with complex TIs or noisy objective functions), parameters obtained from very small gaps are prone

**Table 4** Optimal parameters for quality and time for DS and TIs shown in Fig. 5c and f

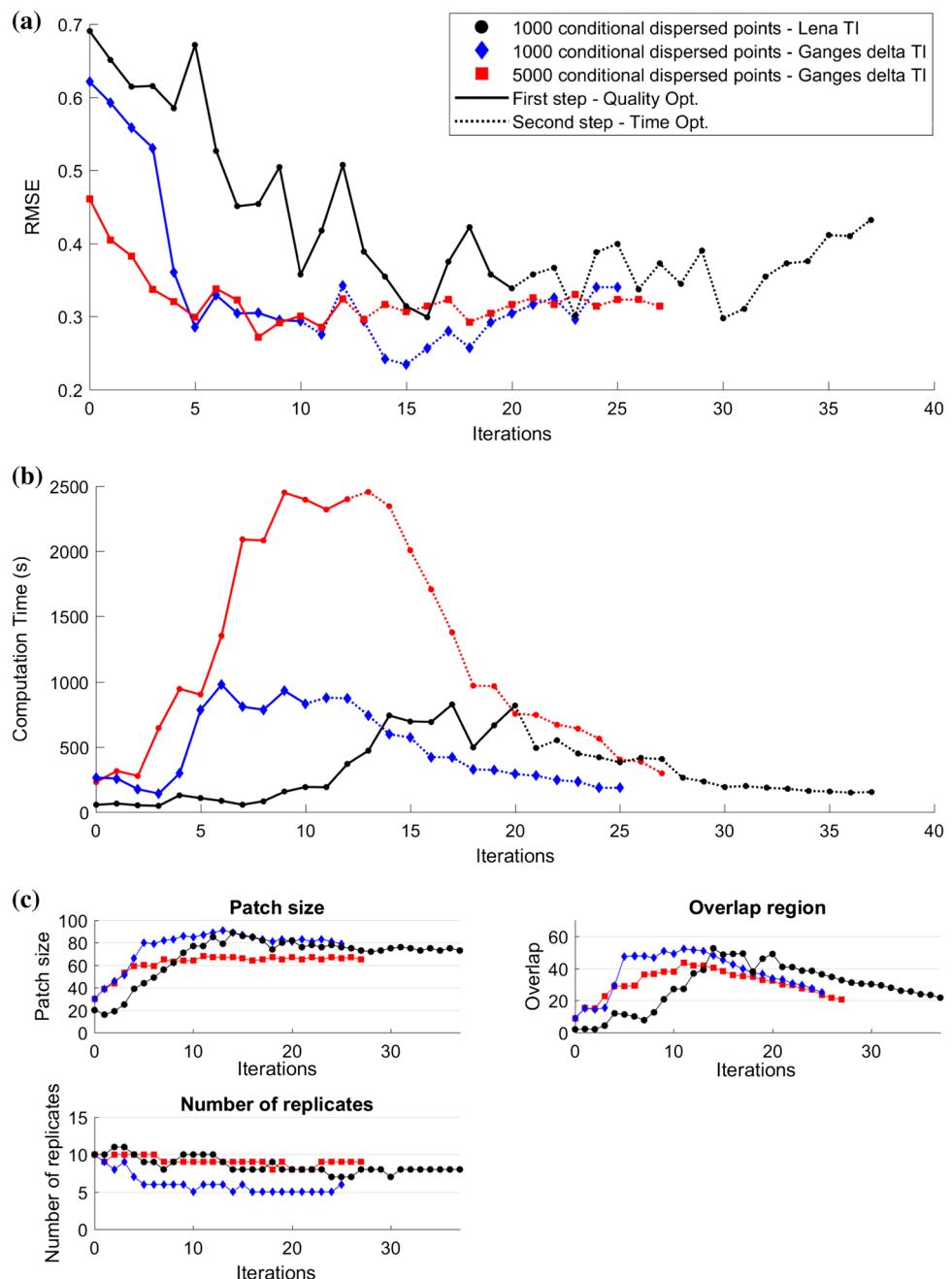
DS parameters	Multivariate TI		Bangladesh TI	
	Quality	Time	Quality	Time
The distance threshold ( $t$ )	0.001	0.051	0.001	0.081
The fraction of scanned TI ( $f$ )	0.34	0.4	0.1	0.14
Number of closest neighbors ( $n$ )	16	11	35	35
Relative weight of variable 1	0.56	0.58	–	–
Relative weight of variable 2	0.31	0.28	–	–
Relative weight of variable 3	0.13	0.14	–	–

**Fig. 9** MPS-APO with different conditional dispersed points performed on IQ with Lena TI (Fig. 5d) and Bangladesh TI (Fig. 5f). **a** Quality reproduction, **b** computational cost and **c** parameters space at each iteration

to miss the larger scale features, which is also undesirable. Our tests show that  $20 \times 20$  gaps can work in most cases considered.

### 3.9 MPS-APO as a tool to compare MPS methods

Although it is not in the scope of the current paper to delve into a comparison of MPS methods, we welcome this opportunity to demonstrate the capabilities of MPS-APO. The MPS-APO algorithm can be used to fairly compare the performance of MPS methods by using their optimal

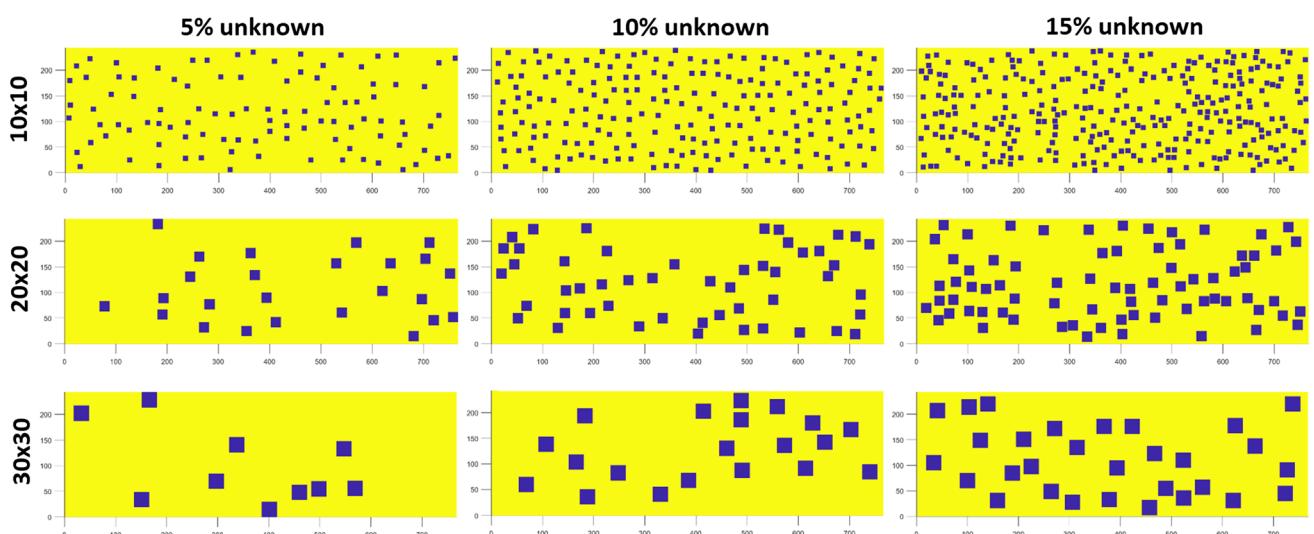


**Table 5** Optimal parameters for quality and time for IQ and TIs shown in Fig. 5d and f

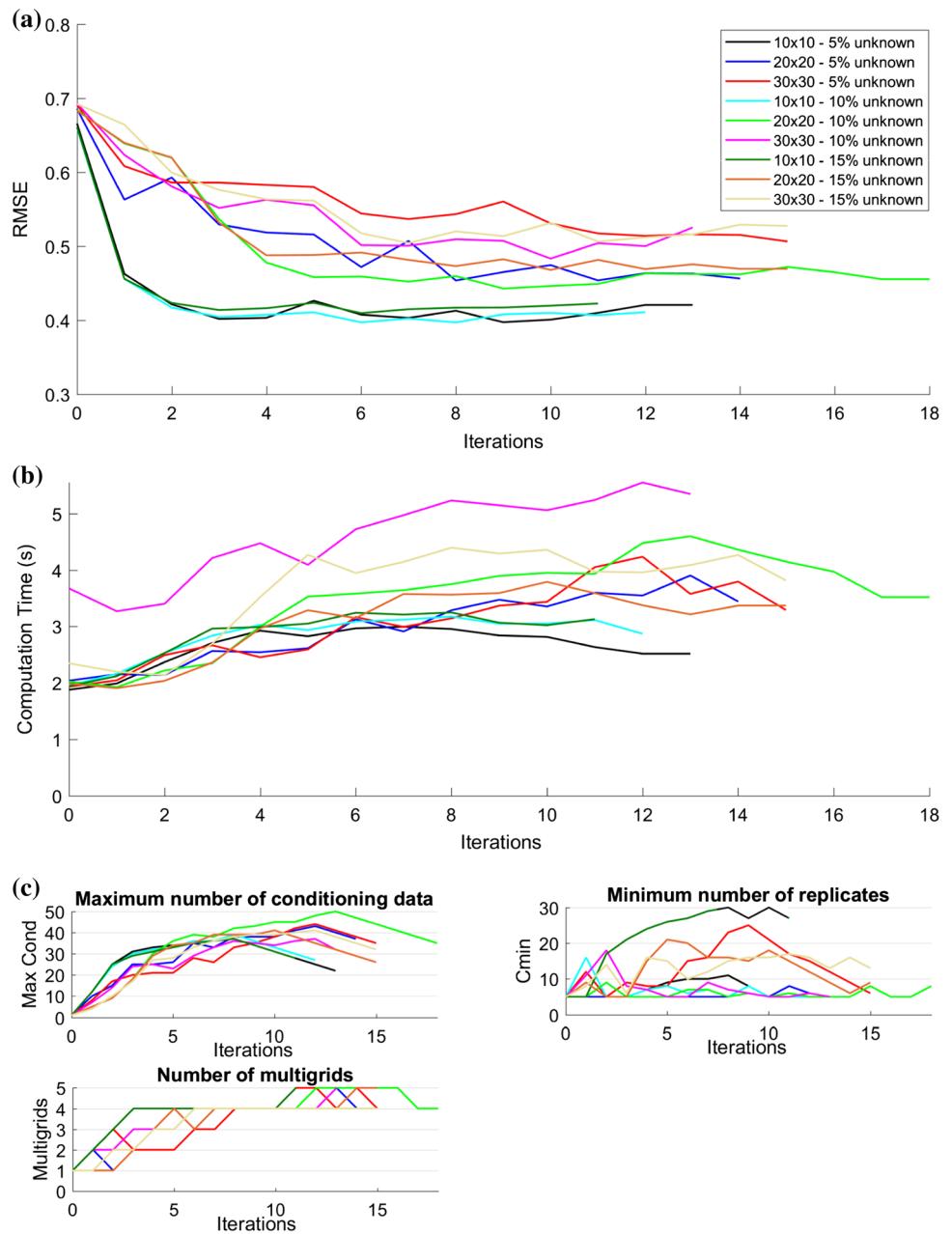
IQ parameters	Lena TI (1000 Cond. Pts)		Bangladesh TI (1000 Cond. Pts)		Bangladesh TI (5000 Cond. Pts)	
	Quality	Time	Quality	Time	Quality	Time
Patch size	82	73	85	79	67	65
Overlap size	46	23	49	27	41	21
Number of replicates	8	8	5	6	9	9

**Table 6** CPU cost of performing MPS-APO for examples of Sects. 3.3–3.6

	SNESIM		FILTERSIM		DS		IQ		
	Marble Thin section TI	Bangladesh TI	Stanford V dataset TI	Bangladesh TI	Multivariate TI	Bangladesh TI	Lena TI (1000 Cond. Pts)	Bangladesh TI (1000 Cond. Pts)	Bangladesh TI (5000 Cond. Pts)
Total MPS-APO time	1736 s	2050 s	33813 s	4477 s	1541 s	1779 s	35573 s	47993 s	104152 s
Average time for each function evaluation	38 s	36 s	451 s	99 s	34 s	30 s	320 s	640 s	1240 s
Number of function evaluations	45	57	75	45	45	60	111	75	84
Number of realizations used in each function evaluation	10	5	10	10	5	5	1	1	1
Average time for each realization	3.8 s	7.2 s	45.1 s	9.9 s	6.9 s	6 s	320 s	640 s	1240 s
Time for one realization with final optimal parameters	2.3 s	3.5 s	44 s	8.3 s	7 s	2.9 s	149 s	187.1 s	300 s

**Fig. 10** Different settings of gap sizes and densities

**Fig. 11** Parameter optimization under different settings of gap density



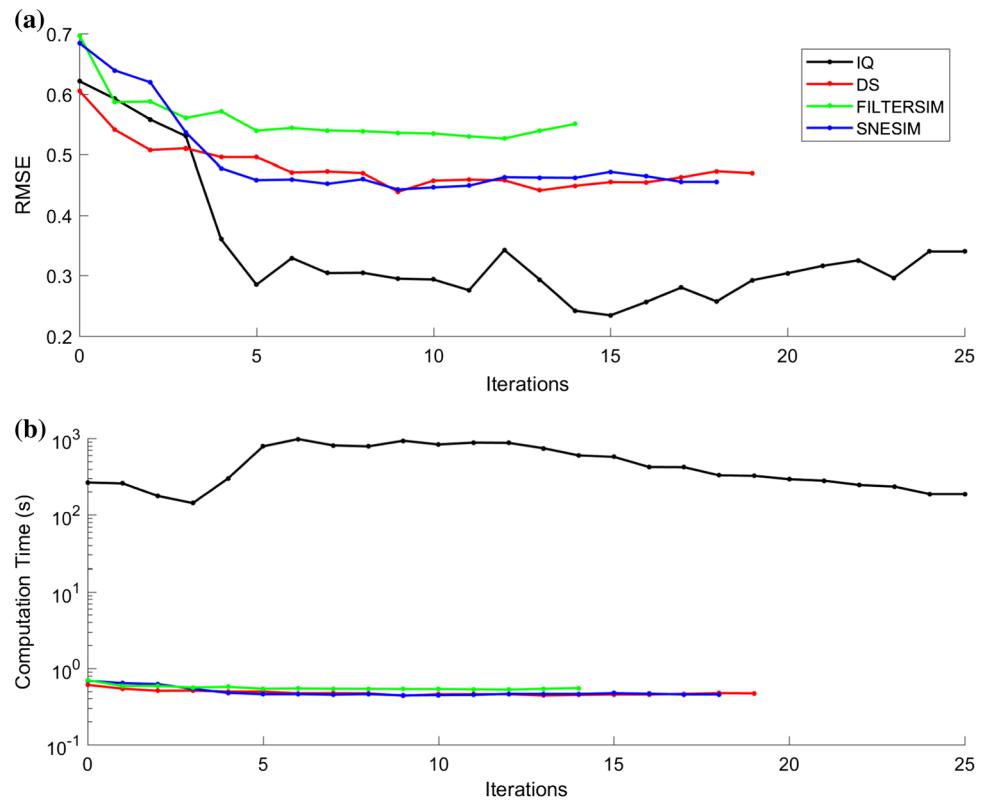
parameters. Since not all methods can work with different kinds of datasets, the TI is chosen on the basis of compatibility with all MPS methods used in this paper. To this end, the binary TI of the Ganges delta (Fig. 5f) was selected. For each MPS method, the MPS-APO is performed on the dataset with 10% unknown values, consisting of 50 square gaps of  $20 \times 20$  pixel. For IQ, 1000 random dispersed points are randomly chosen from the TI to be used as conditioning data. Due to high computational cost of IQ, the objective function only uses one simulation for each iteration in that case.

Figure 6 show the parameter optimization process for SNESIM. The high value for the number of multiple-grids

in Fig. 6c, suggests that the gap approach in MPS-APO is successful in capturing large scale features contained in the TI. The final parameters in Fig. 7 also show the same process for FILTERSIM. However, Fig. 7c shows that in this case the number of multiple-grids are degrading in the last iterations. This happens because in the second step MPS-APO, the parameters are optimized regarding their impact on time instead of quality.

Figure 8 shows the parameterization process for Direct Sampling. Figure 8c suggests the most important parameter regarding the computational time is the threshold: using it to slightly degrade the simulations improves computing time by factor of two. Figure 9 demonstrates the

**Fig. 12** The comparison of MPS method performance in regard to **a** pattern quality reproduction and **b** computational cost



parameterization process for image quilting. In Fig. 9a, the RMSE is almost constant for 20 iteration starting from iteration 5, however Fig. 9b clearly shows that the initial parameters come with an important CPU cost. Because of the time optimization, the final parameters have the same RMSE while the CPU cost is 5 times lower, once again showing the importance of parametrization.

Figure 12 compares these four methods in regards to their ability to reproduce the patterns and the computational cost of doing so. Figure 12a clearly shows that considering the quality of simulations, the IQ algorithm comes first. The results for SNESIM and DS are very close in second place, leaving FILTERSIM in the last place. However in term of CPU cost the results are different: SNESIM and DS are in the first place, followed very closely by FILTERSIM, then IQ comes last.

### 3.10 Global optimum vs local optimum

When the importance of the parameters is different, it may result in different parameter values in different optimization runs for the less sensitive parameters (e.g. In the DS algorithm, the number of neighborhood and threshold are more sensitive than the fraction of TI scanned). Figure 13 presents such an example, where two DS automatic parameterizations are identified on the TI of Fig. 5e. As the figure indicates, both optimizations converged to optimal

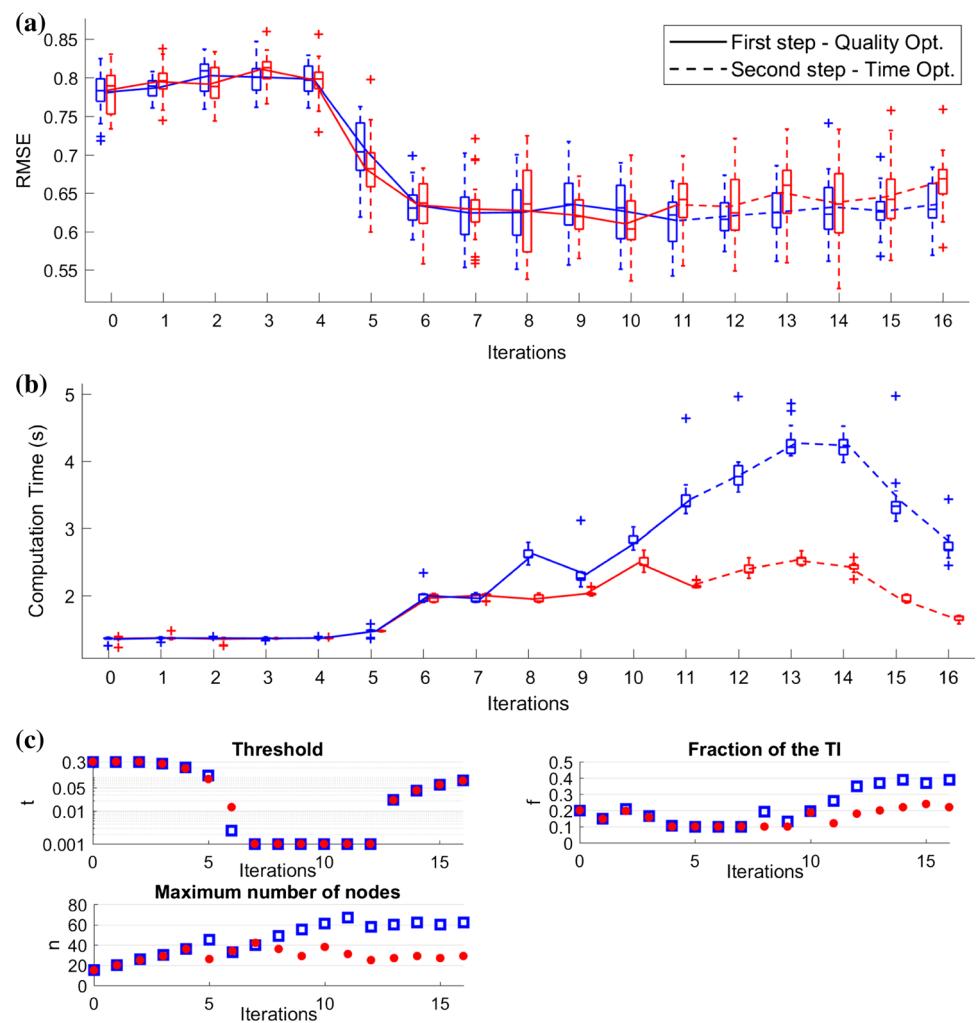
parameters with almost the same RMSE. For DS, the threshold is the most important parameter that controls the quality of the simulations. Although the threshold is exactly the same for the both optimizations, they converged to different parameters for number of nodes and fraction of the TI, resulting in a slightly higher CPU cost. Here it is interesting to note that both solutions can be considered optimal given the stochastic nature of the objective function.

## 4 Conclusion

In this study, MPS-APO is performed on four different classes of geostatistical methods: SNESIM, FILTERSIM, Direct Sampling and Image Quilting. The algorithm is able to automatically approximate the optimal parameters for different types of TIs. We show that MPS-APO is a useful heuristic to automatically approximate optimal parameters that offer good patterns reproduction with minimal computational cost. MPS-APO can also increase the usability of multiple-point statistics for practical applications and especially for non-expert users of MPS algorithms.

Classically, MPS parameters are determined by cumbersome sensitivity analysis, which is not practical when many parameters are present. Furthermore, the sensitivity analysis has to be redone if the setting changes (e.g.,

**Fig. 13** an example of two optimizations (differentiated by color) with identical initial settings, which result in different final parameters



change of TI). MPS-APO can rapidly find one local minimum, however there is always a trade-off between simulation quality and CPU cost. It allows finding justifiable parameters in relation to both quality of pattern reproduction and computational time. While there is no guarantee that the identified parameters correspond to the global optimum, the solution generally provides a good enough solution for practical purposes, in a reasonable time (i.e., much faster than a complete sensitivity analysis).

By standardizing the parameter selection for any simulation algorithm, MPS-APO paves the way for objectively comparing MPS methods under optimal conditions. Indeed, a fair comparison of MPS algorithms can be difficult. Even when undertaking a direct, side-by-side comparison, the choice of good input parameters for each compared approach is subjective. As a consequence, using suboptimal parameters may lead to biased comparisons, which can be prevented by using our approach.

Because of SPSA's efficiency, MPS-APO can reach convergence with a relatively low computational cost. Using parallelization could further improve its speed,

which is the topic of future work. Another avenue is to design alternative objective functions that aim at maximizing the variability between realizations while keeping an optimal pattern reproduction. While these objectives can be contradictory, the problem could be addressed by using multi-objective optimization to find a set of Pareto optimal solutions.

**Acknowledgements** We thank Mathieu Gravey for his insightful comments and extraordinary technical support. We are also grateful for the constructive comments of two anonymous reviewers which helped improving the overall quality of the manuscript.

## References

- Abdulsadda AT, Iqbal K (2011) An improved SPSA algorithm for system identification using fuzzy rules for training neural networks. *Int J Autom Comput* 8:333
- Arpat GB, Caers J (2007) Conditional simulation with patterns. *Math Geol* 39:177–203

- Bai H, Ge Y, Mariethoz G (2016) Utilizing spatial association analysis to determine the number of multiple grids for multiple-point statistics. *Spat Stat* 17:83–104
- Boisvert JB, Pyrcz MJ, Deutsch CV (2007) Multiple-point statistics for training image selection. *Nat Resour Res* 16:313–321
- Boisvert JB, Pyrcz MJ, Deutsch CV (2010) Multiple point metrics to assess categorical variable models. *Nat Resour Res* 19:165–175
- Chiles J, Delfiner P (1999) Geostatistics: Modeling spatial uncertainty. Wiley series in probability and statistics. Wiley, New York
- Dagasan Y, Renard P, Straubhaar J, Erten O, Topal E (2018) Automatic parameter tuning of multiple-point statistical simulations for lateritic bauxite deposits. *Minerals* 8:220
- Dagasan Y, Erten O, Renard P, Straubhaar J, Topal E (2019) Multiple-point statistical simulation of the ore boundaries for a lateritic bauxite deposit. *Stoch Environ Res Risk Assess* 33:865–878
- de Carvalho PRM, da Costa JFCL, Rasera LG, Varella LES (2017) Geostatistical facies simulation with geometric patterns of a petroleum reservoir. *Stoch Environ Res Risk Assess* 31:1805–1822
- Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the 6th international symposium on micro machine and human science. MHS'95, 1995. IEEE, pp 39–43
- Gardet C, Le Ravalec M, Gloaguen E (2016) Pattern-based conditional simulation with a raster path: a few techniques to make it more efficient. *Stoch Environ Res Risk Assess* 30:429–446
- Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Boston
- Gosavi A, Ozkaya E, Kahraman AF (2007) Simulation optimization for revenue management of airlines with cancellations and overbooking. *OR Spectr* 29:21–38
- Gravey M, Rasera LG, Mariethoz G (2019) Analogue-based collocation of remote sensing images using textural information. *ISPRS J Photogramm Remote Sens* 147:242–254
- Gu W, Xiang C, Venkatesh Y, Huang D, Lin H (2012) Facial expression recognition using radial encoding of local Gabor features and classifier synthesis. *Pattern Recogn* 45:80–91
- Guardiano FB, Srivastava RM (1993) Multivariate geostatistics: beyond bivariate moments. In: Soares A (ed) Geostatistics Tróia '92, vol 5. Springer, Dordrecht, pp 133–144
- Honarkhah M, Caers J (2010) Stochastic simulation of patterns using distance-based pattern modeling. *Math Geosci* 42:487–517
- Hooke R, Jeeves TA (1961) “Direct search” solution of numerical and statistical problems. *J ACM* 8:212–229
- Jha SK, Mariethoz G, Kelly BF (2013) Bathymetry fusion using multiple-point geostatistics: novelty and challenges in representing non-stationary bedforms. *Environ Model Softw* 50:66–76
- Kalantari S, Abdollahifard MJ (2016) Optimization-based multiple-point geostatistics: a sparse way. *Comput Geosci* 95:85–98
- Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220:671–680
- Kolbjørnsen O, Stien M, Kjønsberg H, Fjellvoll B, Abrahamsen P (2014) Using multiple grids in Markov mesh facies modeling. *Math Geosci* 46:205–225
- Kothandaraman G, Rotea M (2005) Simultaneous-perturbation-stochastic-approximation algorithm for parachute parameter estimation. *J Aircr* 42:1229–1235
- Lee J, Mukerji T (2012) The stanford VI-E reservoir: a synthetic data set for joint seismic-EM time-lapse monitoring algorithms. Paper presented at the 25th annual report: technical report, Stanford Center for Reservoir Forecasting, Stanford University, Stanford, CA
- Liu Y (2006) Using the Snesim program for multiple-point statistical simulation. *Comput Geosci* 32:1544–1563
- Mahmud K, Mariethoz G, Caers J, Tahmasebi P, Baker A (2014) Simulation of earth textures by conditional image quilting. *Water Resour Res* 50:3088–3107
- Mariethoz G, Caers J (2014) Multiple-point geostatistics: stochastic modeling with training images. Wiley, Hoboken
- Mariethoz G, Renard P, Straubhaar J (2010) The direct sampling method to perform multiple-point geostatistical simulations. *Water Resour Res* 46(11):W11536. <https://doi.org/10.1029/2008WR007621>
- Mariethoz G, McCabe MF, Renard P (2012) Spatiotemporal reconstruction of gaps in multivariate fields using the direct sampling approach. *Water Resour Res* 48(11):W10507. <https://doi.org/10.1029/2012WR012115>
- Meerschman E, Pirot G, Mariethoz G, Straubhaar J, Van Meirvenne M, Renard P (2013) A practical guide to performing multiple-point statistical simulations with the direct sampling algorithm. *Comput Geosci* 52:307–324
- Mustapha H, Dimitrakopoulos R (2011) HOSIM: a high-order stochastic simulation algorithm for generating three-dimensional complex geological patterns. *Comput Geosci* 37:1242–1253
- Nelder JA, Mead R (1965) A simplex method for function minimization. *Comput J* 7:308–313
- Oriani F, Mehrotra R, Mariethoz G, Straubhaar J, Sharma A, Renard P (2018) Simulating rainfall time-series: how to account for statistical variability at multiple scales? *Stoch Environ Res Risk Assess* 32:321–340
- Pourfard M, Abdollahifard MJ, Faez K, Motamed SA, Hosseini T (2017) PCTO-SIM: multiple-point geostatistical modeling using parallel conditional texture optimization. *Comput Geosci* 102:116–138
- Remy N, Boucher A, Wu J (2009) Applied geostatistics with SGeMS: a user's guide. Cambridge University Press, Cambridge
- Renard P, Allard D (2013) Connectivity metrics for subsurface flow and transport. *Adv Water Resour* 51:168–196
- Rezaee H, Marcotte D (2017) Integration of multiple soft data sets in MPS thru multinomial logistic regression: a case study of gas hydrates. *Stoch Environ Res Risk Assess* 31:1727–1745
- Rezaee H, Asghari O, Koneshloo M, Ortiz JM (2014) Multiple-point geostatistical simulation of dykes: application at Sungun porphyry copper system, Iran. *Stoch Environ Res Risk Assess* 28:1913–1927
- Rezaee H, Marcotte D, Tahmasebi P, Saucier A (2015) Multiple-point geostatistical simulation using enriched pattern databases. *Stoch Environ Res Risk Assess* 29:893–913
- Sadegh P, Spall JC (1998) Optimal random perturbations for stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans Autom Control* 43:1480–1484
- Saripally I, Caers J (2008) Evaluating data conditioning accuracy of MPS algorithms and the impact on flow modeling. In: 21th SCRF meeting, Stanford University, 2008
- Schittkowski K (1986) NLPQL: a FORTRAN subroutine solving constrained nonlinear programming problems. *Ann Oper Res* 5:485–500
- Spall JC (1987) A stochastic approximation technique for generating maximum likelihood parameter estimates. In: American control conference. IEEE, pp 1161–1167
- Spall JC (1992) Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans Autom Control* 37:332–341
- Spall JC (1998) Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Trans Aerosp Electron Syst* 34:817–823
- Straubhaar J, Renard P, Mariethoz G, Froidevaux R, Besson O (2011) An improved parallel multiple-point algorithm using a list approach. *Math Geosci* 43:305–328

- Straubhaar J, Renard P, Mariethoz G, Chugunova T, Biver P (2019) Fast and interactive editing tools for spatial models. *Math Geosci* 51:109–125
- Strebelle S (2002) Conditional simulation of complex geological structures using multiple-point statistics. *Math Geol* 34:1–21
- Strebelle S, Cavelius C (2014) Solving speed and memory issues in multiple-point statistics simulation program SNESIM. *Math Geosci* 46:171–186
- Tahmasebi P, Hezarkhani A, Sahimi M (2012) Multiple-point geostatistical modeling based on the cross-correlation functions. *Comput Geosci* 16:779–797
- Tan X, Tahmasebi P, Caers J (2014) Comparing training-image based algorithms using an analysis of distance. *Math Geosci* 46:149–169
- Yang L, Hou W, Cui C, Cui J (2016) GOSIM: a multi-scale iterative multiple-point statistics algorithm with global optimization. *Comput Geosci* 89:57–70
- Zhang T, Switzer P, Journel A (2006) Filter-based classification of training image patterns for spatial simulation. *Math Geol* 38:63–80
- Zhang T, Du Y, Huang T, Li X (2015) Reconstruction of porous media using multiple-point statistics with data conditioning. *Stoch Environ Res Risk Assess* 29:727–738
- Zonoobi D, Kassim AA, Venkatesh YV (2011) Gini index as sparsity measure for signal reconstruction from compressive samples. *IEEE J Sel Topics Signal Process* 5:927–932

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.