minerals

MDPI

Article

Automatic Parameter Tuning of Multiple-Point Statistical Simulations for Lateritic Bauxite Deposits

Yasin Dagasan [1,*], Philippe Renard [2], Julien Straubhaar [2], Oktay Erten [1] and Erkan Topal [1]

[1] Department of Mining and Metallurgical Engineering, Western Australian School of Mines, Curtin University, Kalgoorlie 6430, WA, Australia; oktay.erten@outlook.com (O.E.); e.topal@curtin.edu.au (E.T.)

[2] Centre for Hydrogeology and Geothermics, University of Neuchâtel, Rue Emile-Argand 11, 2000 Neuchâtel, Switzerland; philippe.renard@unine.ch (P.R.); julien.straubhaar@unine.ch (J.S.)

* Correspondence: yasin.dagasan@postgrad.curtin.edu.au; Tel.: +61-8-9088-6195

Received: 25 April 2018; Accepted: 18 May 2018; Published: 22 May 2018

Abstract: The application of multiple-point statistics (MPS) in the mining industry is not yet widespread and there are very few applications so far. In this paper, we focus on the problem of algorithmic input parameter selection, which is required to perform MPS simulations. The usual approach for selecting the parameters is to conduct a manual sensitivity analysis by testing a set of parameters and evaluating the resulting simulation qualities. However, carrying out such a sensitivity analysis may require significant time and effort. The purpose of this paper is to propose a novel approach to automate the parameter tuning process. The primary criterion used to select the parameters is the reproduction of the conditioning data patterns in the simulated image. The parameters of the MPS algorithm are obtained by iteratively optimising an objective function with simulated annealing. The objective function quantifies the dissimilarity between the pattern statistics of the conditioning data and the simulation image in two steps: the pattern statistics are first obtained using a smooth histogram method; then, the difference between the histograms is evaluated by computing the Jensen–Shanon divergence. The proposed approach is applied for the simulation of the geological interface (footwall contact) within a laterite-type bauxite mine deposit using the Direct Sampling MPS algorithm. The results point out two main advantages: (1) a faster parameter tuning process and (2) more objective determination of the parameters.

Keywords: multiple-point statistics; MPS; direct sampling; parameter selection; pseudo-histogram; simulated annealing; laterite; bauxite mining; simulations; pattern statistics

1. Introduction

Multiple-point statistics (MPS) allows the simulation of spatial or temporal random functions by reproducing pattern statistics from an exhaustive data set — the training image (TI) — built from conceptual knowledge [1] or borrowed from analogue sites [2,3]. Due to the capabilities and the power of MPS, its use has been rather widespread for simulating complex structures in earth sciences. This fact paved the way for the development of a large number of MPS algorithms and their application to various contexts [1,4–12].

In this paper, we focus on a novel application of MPS: geological modelling of lateritic bauxite deposits. The objective is to simulate the footwall contact topography, which constitutes the base of the exploitable deposit. The reason why MPS could be interesting in this particular case is that several sites have already been mined out, and the topographies of the footwall contact exposed after the mining operations can be considered as analogues to the footwall contact of a future mining area. In addition, the data sets collected from such mines usually include several variables such as exploration boreholes, exhaustive geophysical data, and production control boreholes [13–15]. In order to account for the multivariate nature of the modelling problem and the necessity to simulate a continuous variable, it was decided to use the Direct Sampling (DS) algorithm [10] in this work.

As in all MPS algorithms [16], DS requires some input data such as the training image (TI) and a set of specific input parameters. The choices of the TI and the input parameters significantly influence the simulation results [17]. Assuming that the selection of a TI has already been made, identifying a correct set of input parameters plays a direct role on the quality of the simulations and the spatial uncertainty quantification. One of the most common approaches to determine the MPS parameters is to manually tune them by conducting a sensitivity analysis [10,16,18]. Computation time and simulation quality are the key performance indicators used in the sensitivity analysis [19]. Because these two indicators often counteract each other, the optimality of a set of input parameters is usually assessed based on the balance achieved between them. Applications of such sensitivity analyses can be found in Liu [19], Meerschman et al. [20], Rezaee et al. [21], Huysmans and Dassargues [22].

More generally, the quality of the simulations is often difficult to assess since many different criteria can be used and the parameter identification is based on a manual procedure and a visual inspection of the results. More rigorous methods define a set of indicators to quantify the similarity between the input data and the simulations. Quantification of the similarity between the TI, conditioning data and the resulting simulations is usually based on the reproduction of first and second order statistics. For instance, Meerschman et al. [20] quantify the error between the TI and model statistics based on the calculated errors in connectivity, histograms and variograms. Quantifying the multiple-point statistics error, on the other hand, can be done by multiple-point histograms [18,23,24], cumulants [8], connectivity functions [25], spatial patterns [26], smooth histograms [27], or connectivity indicators [3]. However, carrying out such a sensitivity analysis manually requires running many simulations and is cumbersome and time-consuming.

The specific aim of the present paper is to present a new automated technique to determine the appropriate input parameters of the MPS algorithm for the case of bauxite deposits. Some of the techniques and ideas that are developed in this context can easily be re-used and extended for more general situations (stratified deposits), as we will discuss at the end of the paper. The benefits of an automated method are twofold: it leads to higher quality simulations through the enhancement of pattern reproduction; and it provides less labour intensive and more objective parameter tuning. The approach presented in this paper makes use of a stochastic optimisation framework to automate the parameter tuning process. Input parameters are utilised as the decision variables to minimise an objective function which quantifies the mismatch between the pattern statistics of the conditioning data and the generated realisations. Computing the objective function is performed in two steps: First, the smooth histogram formulation of Melnikova et al. [27] is used to quantify the pattern statistics of the boreholes and that of the simulations. We have selected that approach since it allows working with continuous variables, while other simpler techniques would not allow computing histograms of patterns of continuous variables. Second, the dissimilarity between pattern histograms is evaluated utilising the Jensen-Shannon divergence (JS) [28]. In order to observe the effect of the optimisation algorithm on the tuned parameters, a number of optimisation techniques have been utilised to minimise the objective function. Nevertheless, the Simulated Annealing (SA) algorithm [29] has proven to be the most efficient method amongst the ones tested, as the objective function may contain many local minima. After hundreds of iterations, the SA algorithm converges and provides the parameters yielding the minimum JS divergence value.

The remainder of the paper is organised as follows: Section 2 reviews the underlying methods used to develop the proposed approach, Section 3 presents the setup of the problem using the data from a bauxite mine and implementation of the approach, Section 4 is dedicated to the analyses of the results and Section 5 concludes the key features of the automated parameter tuning process presented.

2. Overview of Underlying Methods

In this section, the methods that are used in the proposed methodology are reviewed. The aim is to provide the required background information to understand the implementation of the methodology described in Section 3.

2.1. Direct Sampling Algorithm

Direct Sampling is a pixel-based MPS algorithm which is used to simulate a random function Z [10]. Being a sequential simulation algorithm, it successively visits all the locations x of a regular simulation grid (SG) and generates simulated $Z^*(x)$ values, until all the grid nodes are informed. Once all the conditioning data (if available) are assigned in the SG, the algorithm follows a predefined random path to visit all the non-informed grid nodes x to perform the simulations. Having chosen the n maximum number of closest grid nodes parameter of the DS, the algorithm finds n number of informed neighbours at grid nodes $\{x_1, x_2, ..., x_n\}$ around x and computes the lag vectors $L = \{h_1, h_2, ..., h_n\}$ to define the data event $d_n(x, L) = \{Z(x + h_1), ..., Z(x + h_n)\}$. The TI is randomly scanned at y locations until the distance between the patterns $d_n(x, L)$ and $d_n(y, L) = \{Z(y + h_1), ..., Z(y + h_n)\}$ falls below a predefined threshold t or until the maximal scan fraction f of the TI is reached. Then, the value $Z(y)$ at the scanned node y is taken as the best match and is pasted on the grid node x of the SG. These steps are repeated until all the nodes in the SG are simulated.

There are different ways of computing the distance between two data events (or patterns) in the DS algorithm and the choice depends mainly on the type of the variable. For instance, the distance for continuous variables can be calculated by using the Manhattan distance:

$$d\{d_n(x, L), d_n(y, L)\} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Z(x_i) - Z(y_i)|}{d_{\max}} \in [0,\ 1] \tag{1}$$

where

$$d_{\max} = \max_{y \in TI} Z(y) - \min_{y \in TI} Z(y) \tag{2}$$

When calculating the distances between the patterns retrieved from the SG and the TI, different weightings can be given to the data event nodes based on their distances $\|h_i\|$ to the central node. This is carried out by a weighting factor α_i applied to each data event node such that:

$$\alpha_i = \frac{\|h_i\|^{-\delta}}{d_{\max}\sum_{j=1}^{n}\|h_j\|^{-\delta}} \tag{3}$$

where δ is the power for computing the weight. A specific weighting w can also be used to achieve pattern consistency in the neighbourhood of the conditioning data. The weighting factor α_i then becomes:

$$\alpha_i = \frac{\beta_i\|h_i\|^{-\delta}}{d_{\max}\sum_{j=1}^{n}\beta_j\|h_j\|^{-\delta}} \tag{4}$$

where

$$\beta_i = \begin{cases} w & \text{if the } i^{th} \text{ node (i.e., } x_i\text{) is a conditioning location (hard data)} \\ 1 & \text{otherwise} \end{cases}$$

The DS algorithm also allows multivariate simulations utilising the multiple-point relationship between the variables in the training data set. The training image TI used in multivariate simulations is comprised of m variables. Considering the variables $Z_1(x), ..., Z_m(x)$ $(k = 1, ..., m)$ used in the multivariate analysis, each variable $Z_k(x)$ may have a different data event $d^k_{n_k}(x, L) = \{Z_k(x + h^k_1), ..., Z_k(x + h^k_{n_k})\}$

with different L_k lag vectors. The joint data event $d_n(x) = \{d^1_{n_1}(x, L^1), ..., d^m_{n_m}(x, L^m)\}$ is then used to scan the TI to find a compatible pattern.

In the original version of the DS [10], the algorithm combines all the specific distances using a weighted average. However, in this work, we use a different implementation of the DS called DeeSse [30]. Instead of using a global threshold and a set of weights, DeeSse uses a variable specific threshold t_k for each variable Z_k. The TI is scanned until the distance for each pattern is below the corresponding threshold, or the maximum scan fraction is reached [30,31]. More detailed information on the implementation and application of the DS algorithm can be found in Mariethoz et al. [10] and Meerschman et al. [20].

2.2. Comparing Patterns with Smooth Histograms

The smooth histogram is a pseudo-histogram reflecting the pattern statistics of an image [27]. Since it is based on the pixel values rather than the pattern counts, it can be computed for both discrete and continuous images. Melnikova et al. [27] use such a histogram to compare the multiple-point statistics of a continuous model with a discrete training image for an inverse problem. The comparison is carried out by first defining a search template T to collect the unique patterns $pat_j^{TI,un}$ of the categorical training image. These patterns are then used to construct the categories (classes) of the pseudo-histograms $H^{d,m}$ and $H^{d,TI}$ of the model image and the TI, respectively. Because the categories of the constructed pseudo-histograms are discrete, any pattern pat_i^m observed in the continuous model image m would not fall into one of these categories. Instead, it contributes to all $N^{TI,un}$ numbers of unique categories with a value between 0 and 1. This value is calculated based on the similarity of the patterns in the histogram categories and the patterns of the model image. The jth bin of the model histogram $H^{d,m}$ is then computed by:

$$H_j^{d,m} = \sum_{i=1}^{N^m} \frac{1}{(1 + A \left\| pat_i^m - pat_j^{TI,un} \right\|_2^k)^s} \tag{5}$$

Similarly, the pseudo-histogram of the TI can be computed by calculating the patterns of the categories and all the patterns in the TI, as in the following:

$$H_j^{d,TI} = \sum_{i=1}^{N^{TI}} \frac{1}{(1 + A \left\| pat_i^{TI} - pat_j^{TI,un} \right\|_2^k)^s} \tag{6}$$

where A, k and s are user-defined parameters to shape the pattern similarity function. Melnikova et al. [27] state that these parameters not only define how well the pseudo-histogram approximates the true frequency distribution, but they also control the degree of smoothing. However, there is a trade-off between the degree of smoothing achieved and the true frequency distribution approximated. Therefore, optimal values balancing these are required. Melnikova et al. [27] define $A = 100$, $k = 2$ and $s = 2$ as optimal parameters balancing this trade-off.

The comparison of two pseudo-histograms requires a dissimilarity metric. In our case, we chose the Jensen–Shannon (JS) divergence to calculate the dissimilarity of two pseudo-histograms. The JS is used to quantify the dissimilarity between two density distributions p and q by averaging the Kullback–Leibler divergences, as in the following [32]:

$$d_{JS}(p, q) = \frac{1}{2}\sum_i p_i \log\left(\frac{p_i}{q_i}\right) + \frac{1}{2}\sum_i q_i \log\left(\frac{q_i}{p_i}\right) \tag{7}$$

where p_i and q_i represent the probability densities at the ith bins. Calculation of the dissimilarity also requires the pseudo-histograms to share the same (mutual) classes.

2.3. Generalised Simulated Annealing

Simulated Annealing (SA) is a stochastic optimisation tool to solve complex optimisation problems by mimicking the annealing process of a molten metal [33,34]. Being inspired by the Metropolis algorithm [35], it was first proposed by Kirkpatrick et al. [29] to find the global optimum of a complex objective function. The method approximates the global minimum by exploring the solution space at finite locations. The combination of stochastic exploration and a cooling scheme controls the probability of accepting worse solutions and avoiding remaining trapped in local minima. The artificial temperature is high in the initial stages of the optimisation, therefore, worse solutions have a higher probability of acceptance. As the optimisation progresses, the temperature is lowered and the focus is shifted toward accepting only better solutions to identify the minimum more accurately.

Given an objective function $f(x)$ such that $x = (x_1, x_2, ..., x_n)$, a standard SA algorithm uses the following steps to find the global minimum [36]:

1. Choose a high initial temperature T_0 value, an initial solution x^0 and evaluate the objective function, $E^0 = f(x^0)$.
2. Propose a new solution x^{i+1}:
   - Generate a candidate solution x^{i+1} from the current one (x^i) using a predefined visiting distribution.
   - Evaluate the change in the objective function for the candidate solution, $\Delta = f(x^{i+1}) - f(x^i)$.
   - Accept the iteration if it reduces the objective function, $\Delta < 0$.
   - Otherwise, accept or reject it based on a probability of acceptance criterion.
3. Repeat step 2 for L number of iteration times keeping T constant.
4. Reduce the temperature to T_{n+1} using a cooling function.
5. Repeat steps 2–4 until the convergence criteria is satisfied.

The choice of the visiting distribution in step 2 has a significant effect on the efficiency of the SA algorithm. Therefore, different visiting distributions for the SA have been investigated by some researchers such as Szu and Hartley [37] and Tsallis and Stariolo [38]. In this research, the Generalised Simulated Annealing algorithm (GSA) [38] with the distorted Cauchy–Lorentz distribution is used. As for the stopping criteria, the GSA offers different options such as maximum running time, maximum function calls, maximum iteration number or a threshold value for the objective function.

3. Problem Setup and Methodology

Before describing the details of the proposed methodology, let us present the problem more precisely. The ultimate aim is to determine the MPS parameters to simulate the interface between the ore and waste within a laterite-type bauxite deposit.

The data to perform the simulations comes from two mine sites. The first one (Figure 1a,b) comprises a finite set of depth measurements collected from regularly spaced boreholes (conditioning data and primary variable) and an exhaustive Ground Penetrating Radar (GPR) survey (soft data and secondary variable). The position of the interface in this mine site is aimed to be simulated by DS, using these data sets as conditioning information. The second site is a mined-out analogue area providing a training data set. It includes the topographical survey of the exposed deposit base (primary variable) as well as a GPR survey done before the mining operation. This training data set, which is used as a bivariate training image, can be seen in Figure 1.

In addition to this data, the simulation of the interface with DS requires the input parameters (number of neighbours, distance thresholds, etc.) to be determined. The aim of the proposed methodology is, therefore, to automatically obtain an optimal set of parameters for that purpose.

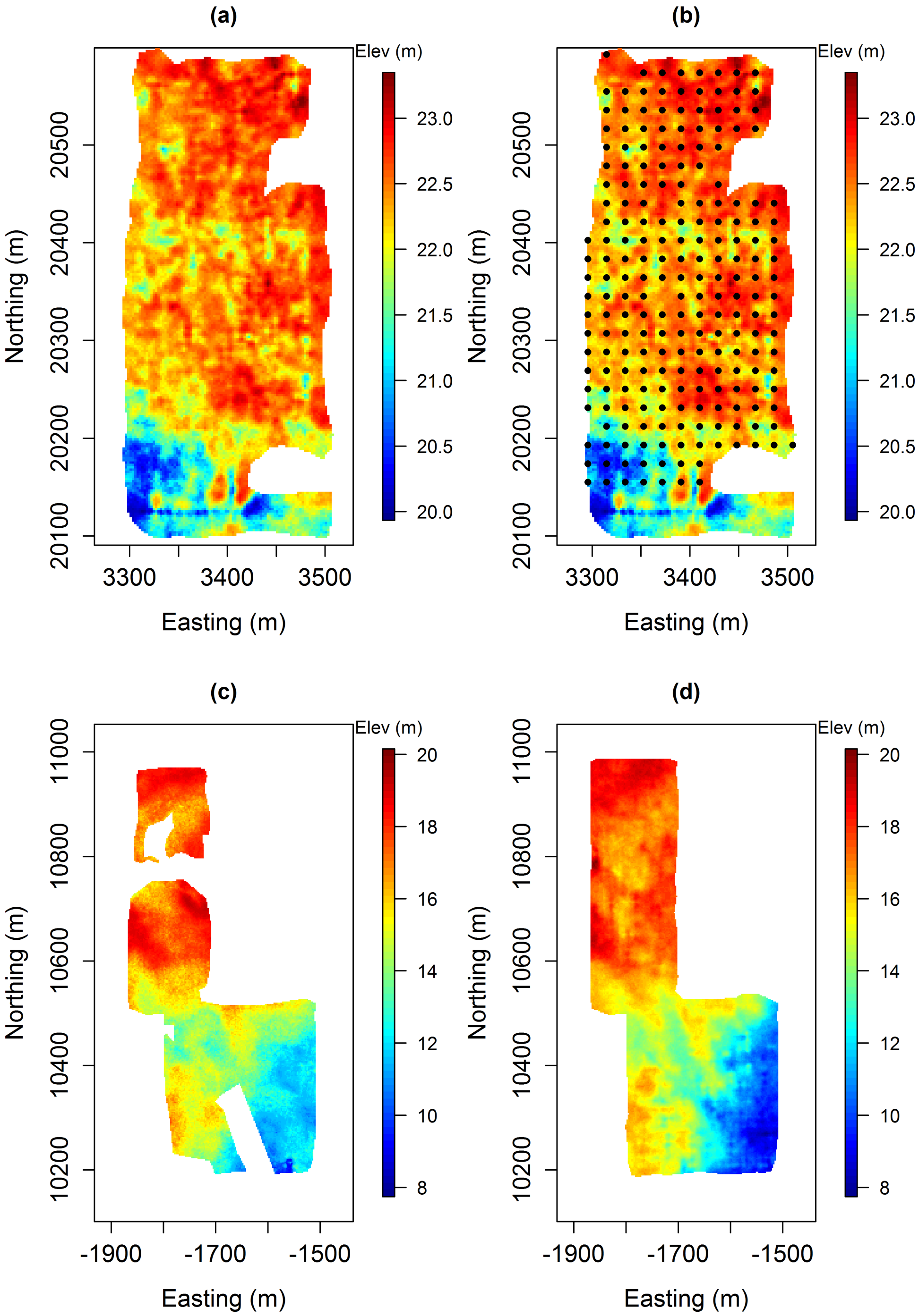

Figure 1. Simulation and the TI variables: (**a**) GPR variable used as the secondary simulation variable; (**b**) borehole data used to condition the simulations (primary variable-black dots overlying the GPR survey); (**c**) floor survey of a previously extracted mine area (primary TI variable) and (**d**) GPR data of the previously mined area collected prior to the extraction (Secondary TI variable).

In lateritic bauxite deposits, the data sets may be dense and regularly organised. This offers the possibility to define a specific quality metric in which the patterns observed in the conditioning data can be taken as the target. Therefore, the general principle of the proposed methodology is to minimise an objective function representing the dissimilarity of the conditioning data with the simulated image in terms of pattern statistics. The calculation of the dissimilarity requires the pattern statistics to be computed. This is only done once for the conditioning data. For the simulated image, however, the pattern statistics change with each perturbation of the parameters and needs to be updated.

Methodology

The computation of the conditioning data pattern statistics first requires the migration of the borehole datapoints into the SG. This step is needed to convert the punctual data into the gridded type. Once this step is done, a search template T is defined to retrieve the patterns contained in the borehole data. Since we are comparing the pseudo-histograms of one fully informed simulation image and a partially-informed borehole data image, a special search template needs to be constructed. Therefore, the construction of the search template T is based on two parameters: (1) the number of grids between borehole data and (2) the number of borehole data desired to be used in one category of the pseudo-histogram. This is illustrated in the following example: say we have borehole data located in a grid at every five nodes in the x and y directions. If we want to capture a four boreholes data pattern, we choose the size of the search template to be 5 × 5. The constructed search template scans the borehole data grid and collects the patterns once four borehole data is captured by the search template. The collected patterns serve as the categories of the pseudo-histogram to be constructed. In order to account for the non-stationarity, the means $\overline{pat_i^C}$ are subtracted from each pat_i^C, focusing on the variation of each pattern around its mean. The steps mentioned above are illustrated in Figure 2.

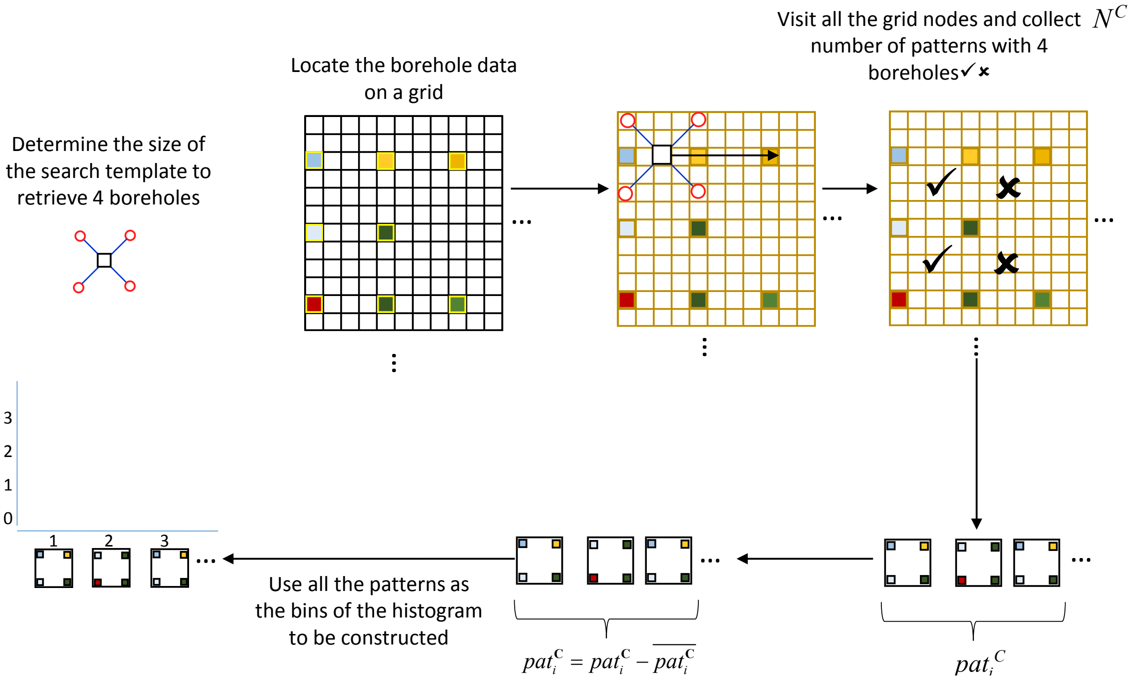
Determine the size of the search template to retrieve 4 boreholes
Locate the borehole data on a grid
Visit all the grid nodes and collect N^C number of patterns with 4 boreholes ✓ ×
3
2
1
0
1
2
3
Use all the patterns as the bins of the histogram to be constructed
$pat_i^C = pat_i^C - \overline{pat_i^C}$
pat_i^C

Figure 2. Construction of the smooth histogram categories of the of borehole data.

Following the construction of the smooth histogram categories for the conditioning data, the pseudo-histogram $H^{d,C}$ for the conditioning data is computed by the following:

$$H_j^{d,C} = \sum_{i=1}^{N^C} \frac{1}{(1 + A \left\| pat_i^C - pat_j^C \right\|_2^k)^s} \tag{8}$$

where N^C represents the number of patterns captured from the SG.

Once the smooth histogram of the borehole data is generated, the SA algorithm runs the DS algorithm to produce a realisation. The smooth histogram of this realisation is then computed using the categories created for the conditioning data histogram. In order to calculate the contribution of each pattern pat_i^{SIM} in the histogram categories $H_j^{d,SIM}$, the means $\overline{pat_i^{SIM}}$ are again subtracted from each pat_i^{SIM}. The pseudo-histogram $H^{d,SIM}$ of a realisation is calculated as follows:

$$H_j^{d,SIM} = \sum_{i=1}^{N^{SIM}} \frac{1}{(1 + A \left\| pat_i^{SIM} - pat_j^{C} \right\|_2^{k})^{s}} \tag{9}$$

where N^{SIM} is the number of patterns captured in the realisation image generated. Because there are more patterns in the simulation image than the conditioning data image, the weights in each category of the pseudo-histograms of the realisations are expected to be higher than those of the conditioning pseudo-histogram. In order to compare the pseudo-histograms, they are normalised before the computation of the Jensen–Shannon divergence. The normalisation is carried out through the following:

$$H_j^{d}{}^{Nor} = \frac{H_j^{d} - 1}{\sum\limits_{i=1}^{N^C} (H_i^{d} - 1)} \tag{10}$$

where $H_j^{d}{}^{Nor}$ is the normalised weight corresponding to the weight of the j^{th} category H_j^d of either $H_j^{d,SIM}$ or $H_j^{d,C}$. Subtraction of 1 from each individual pseudo-histogram category has been done to remove the weight contribution of each pattern $H_j^{d,C}$ in its own j^{th} category. The Jensen-Shannon divergence is then calculated as follows:

$$O = d_{JS}(H_j^{d,C_{Nor}}, H_j^{d,SIM_{Nor}}) = \frac{1}{2}\sum_j H_j^{d,C_{Nor}} \log\left(\frac{H_j^{d,C_{Nor}}}{H_j^{d,SIM_{Nor}}}\right) + \frac{1}{2}\sum_j H_j^{d,SIM_{Nor}} \log\left(\frac{H_j^{d,SIM_{Nor}}}{H_j^{d,C_{Nor}}}\right) \tag{11}$$

Starting from an initial set of x^0 DeeSse parameters (vector of decision variables), the SA algorithm runs the DS algorithm to produce a realisation and evaluate the objective function O_1. Then, the parameters are iteratively and randomly perturbed using the Cauchy-Lorentz visiting distribution. A new value of the objective function O_{i+1} is computed and the new set of parameters x_{i+1} is accepted or rejected according to a probability acquired from the generalised Metropolis algorithm. Factors influencing the probability of acceptance comprise the previous and candidate objective function values (O_i, O_{i+1}) as well as the number of iterations that have been made so far. During the initial iterations, the SA has a relatively high probability of accepting changes that may worsen the objective function. In order to explore the function that we want to minimise later in the process, this probability is reduced allowing the identification of the optimum better. The SA algorithm runs until the convergence is achieved and the objective function O is minimised. The overall flowchart of the methodology is shown in Figure 3.

The methodology was mainly implemented using R statistics software [39]. The MPS simulations were performed by calling the DeeSse algorithm (which is coded in C) from R. For the optimisation part, we implemented the GSA using the GenSA package of R software [40].

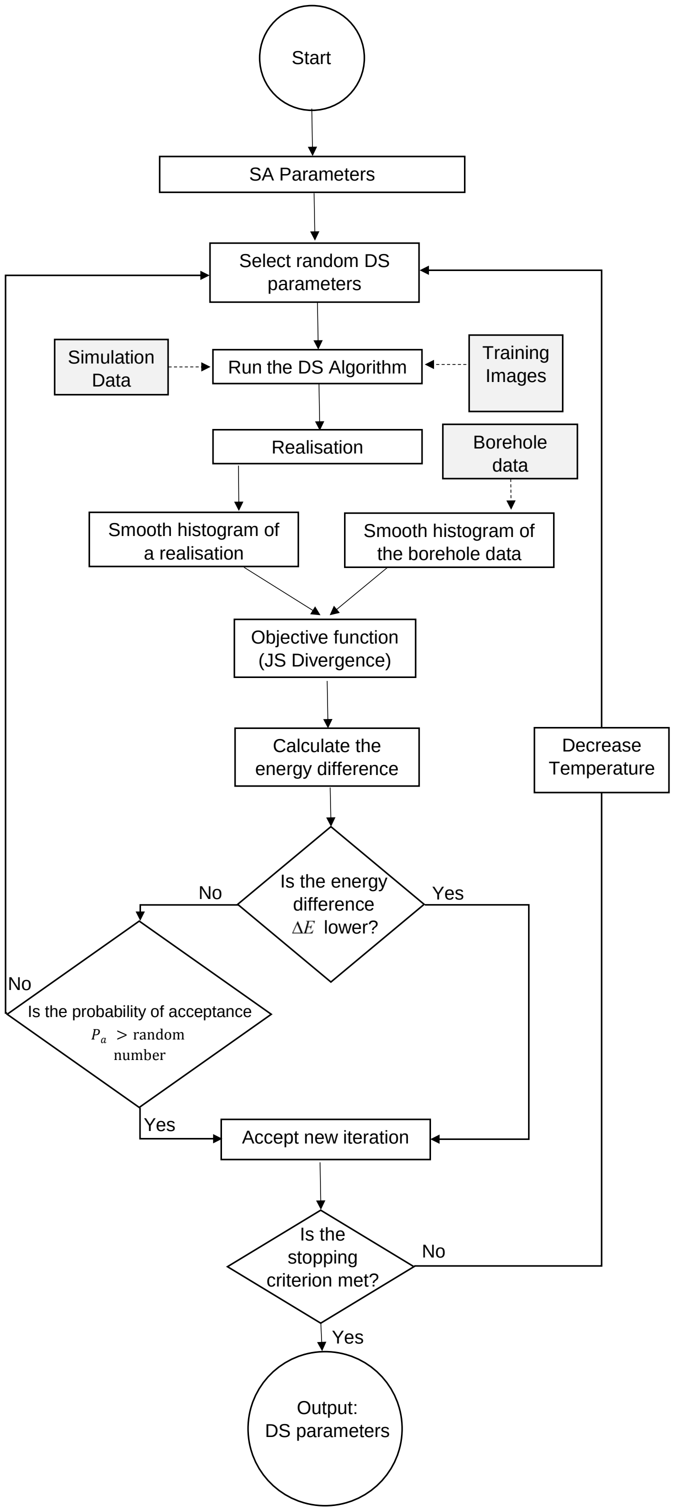
Start
SA Parameters
Select random DS parameters
Simulation Data
Run the DS Algorithm
Training Images
Realisation
Borehole data
Smooth histogram of a realisation
Smooth histogram of the borehole data
Objective function (JS Divergence)
Calculate the energy difference
Decrease Temperature
No
Is the energy difference ΔE lower?
Yes
No
Is the probability of acceptance P_a > random number
Yes
Accept new iteration
Is the stopping criterion met?
No
Yes
Output: DS parameters

Figure 3. Flowchart of the methodology.

4. Results and Discussion

4.1. Implementation and Analysis of the Tuned Parameters on the Simulations

As explained in Section 3, the aim is to model the lateral variability of the ore/waste contact surface in a lateritic bauxite deposit using borehole and GPR datasets (Figure 1). For that purpose, we use a training dataset (bivariate TI) coming from an analogue site (Figure 1c,d). The simulation grid has a size of 97 and 214 nodes in the x and y directions, respectively. The TI, on the other hand, is 180 × 400 in grid size. The spacing between the nodes of both the SG and the TI grids are 2.38 m. Since the data sets have apparent drifts, the types and the coefficients of these drifts were initially detected. First order trend surfaces for the TI and the second order trend surfaces for the simulation variables were found suitable and subtracted from the data sets to obtain the residuals. The residuals obtained were then used to perform the MPS simulations. Following the simulations, the trend surfaces were added back to each realisation.

Several combinations of DS parameters were initially tested to determine the appropriate simulation parameters by visually checking the simulation qualities. These initial attempts have shown that the simulation qualities were rather sensitive to changes in the parameter, making it difficult to identify optimal parameters through visually analysing the resulting simulations. The proposed methodology was then implemented. Five parameters that were considered during the optimisation procedure were w_{hd}, n_{hd}, n_{GPR}, t_{hd} and t_{GPR}. The weighting factor w_{GPR} of the GPR data was not selected for the optimisation, as the simulation grid SG_{GPR} of the GPR data was already exhaustively informed and any change in w_{GPR} would not affect the simulations. One of the main parameters of the DS algorithm, the scan fraction f, was also not considered in the procedure. Based our previous trials, it did not affect the simulation quality significantly for this example. This observation of ours is rather consistent with the results presented in Meerschman et al. [20]; they state that the parameter f has a small effect on the simulation qualities unless it is below 0.2 for continuous simulations. Therefore, the scan fraction f was kept equal to 0.5 throughout the optimisation procedure.

As the first step of our proposed algorithm, a search template is built to capture 4 points to construct the pseudo-histogram. Because the boreholes are drilled on a regular grid of 19.05 × 19.05 m and the grid spacing is 2.38 m, they are located in the SG at every 9th node in the x and y directions. Therefore, a 9 × 9 search template was constructed to capture four borehole data at a time. This search template was then used to construct the pseudo-histogram $H^{d,C}$ of the borehole patterns.

Having constructed the pseudo-histogram of the conditioning data, the SA algorithm was run with the following initial parameters: $w_{hd} = 10$, $n_{hd} = 5$, $n_{GPR} = 20$, $t_{hd} = 0.01$ and $t_{GPR} = 0.1$. These parameters were selected as the ones yielding visually good simulation quality during the manual sensitivity analyses. In addition, the optimisation is also constrained by an upper and lower boundary, as can be seen in Table 1. The resulting realisation is shown in Figure 4a.

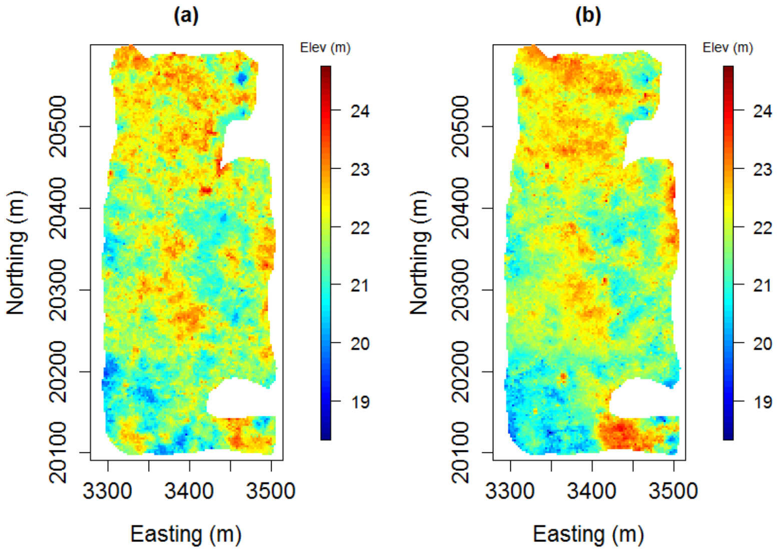

Figure 4. Realisations produced with (**a**) the initial parameters and (**b**) final parameters.

The pseudo-histogram $H^{d,SIM}$ of the initial realisation is then calculated to compute the d_{JS} mismatch. The JS divergence d_{JS} between the pseudo-histograms $H^{d,C}$ and $H^{d,SIM}$ was initially calculated as 0.102. Using 30,000 function calls as the stopping criteria, the SA algorithm converged and found the parameters for the DS algorithm yielding a local minimum divergence value. The resulting realisation can be seen in Figure 4b. The parameters along with their associated JS divergence values are provided in Table 1.

Table 1. Initial and optimum parameters together with the associated JS divergence values.

Parameters	w_{hd}	n_{hd}	n_{GPR}	t_{hd}	t_{GPR}	d_{JS}
Lower Boundaries	1	1	1	0.0001	0.0001	0.1520
Initial Parameters	10	5	20	0.010	0.10	0.1020
Automatic Tuning	13.07	14	45	0.009	0.305	0.0667
Upper Boundaries	15	50	50	0.5	0.5	0.1793

Although the initial and final realisations exhibit a noticeable difference, the average of 40 realisations generated using the initial and optimum parameters do not show an apparent dissimilarity, as can be seen in Figure 5. However, the interquartile ranges of the elevation values calculated at each grid node noticeably dropped after the parameter tuning procedure, as illustrated in Figure 5.

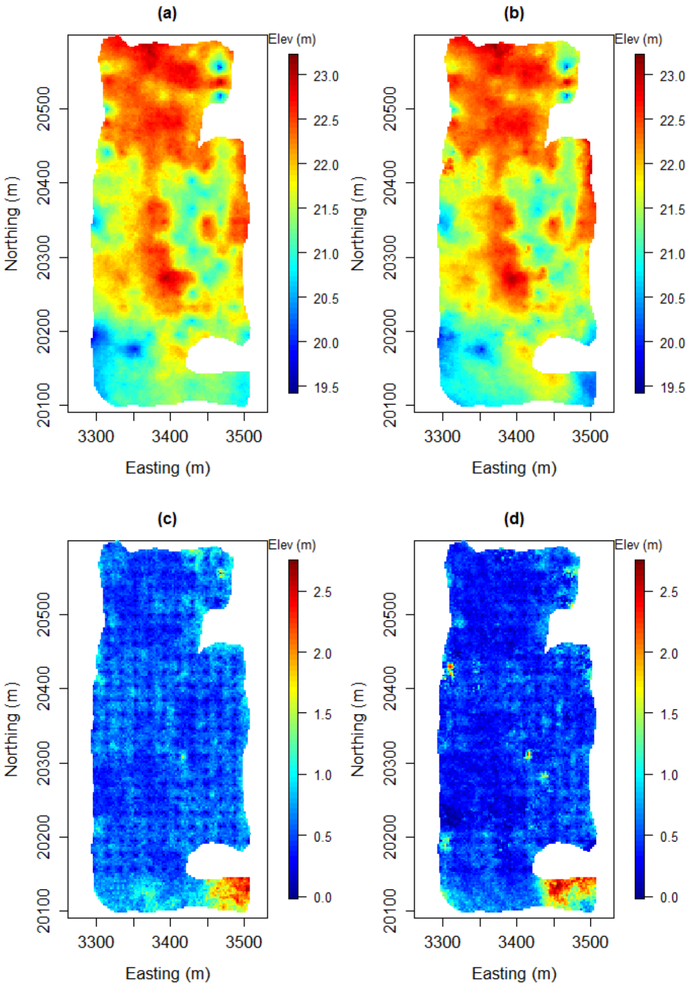

Figure 5. (**a**) Initial average of the realisations; (**b**) final average of the realisations. Interquartile range maps: (**c**) before the parameter optimisation and (**d**) after the parameter optimisation.

This can be seen more clearly in the distributions of IQR values shown in Figure 6. Similarly, the evolution of the variograms in Figure 7 also shows that the variability between the realisations has decreased. In other words, the statistical fluctuations are better centred around the experimental variogram of the borehole data. These results indicate that although the means of the realisations exhibit a considerable similarity, there is a reduction in the estimated uncertainty around the mean. Since the optimised realisations are richer in terms of conditioning data patterns, the model of uncertainty would represent the original data variability better. Therefore, we consider the resulting uncertainty reduction as an improvement.

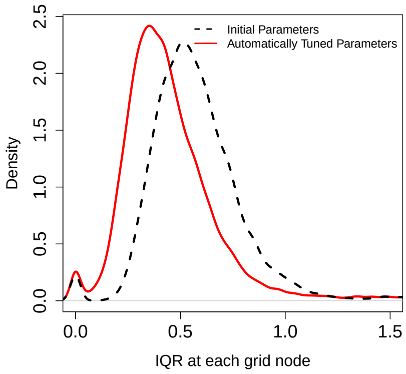

Figure 6. Evolution of the distributions of the IQR values calculated at each grid node. Mean of the IQR values dropped from 0.60 to 0.47 in automatically found parameters.

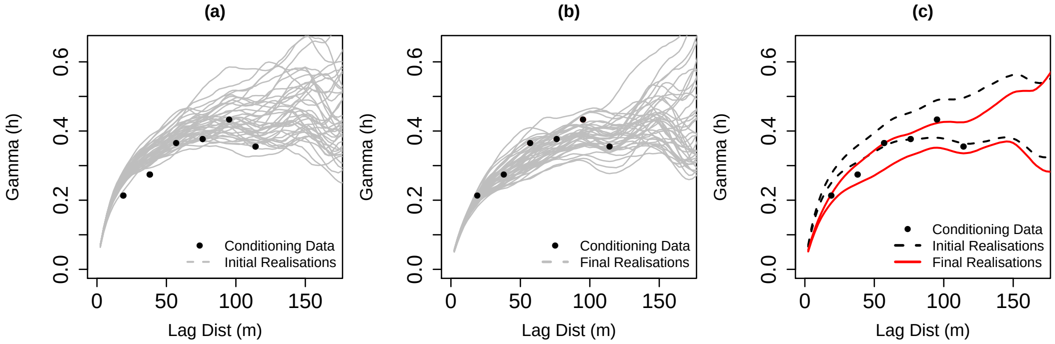

Figure 7. Variograms of the realisations and the experimental variogram of the borehole data: (**a**) Variograms before the optimisation; (**b**) variograms after the optimisation and (**c**) 99.9% confidence interval of the variograms before and after the optimisation.

4.2. *Effect of the Optimisation Method and the Initial Parameters on the Final Results*

In order to test the choice of the SA, we compared the performances of different optimisation techniques. The parameter tuning process has also been carried out using other optimisation methods. These optimisers include the L-BFGS (quasi-newton type) [41] and BOBYQA (trust region) [42] methods. The optimisers were set to run with different initial parameters. These included a set of input parameters yielding a high JS divergence value, suggested parameters in Meerschman et al. [20] and our initial parameters based on the manual sensitivity analysis.

The results in Figure 8 indicate that the final value of the JS divergence is highly dependent on the initial parameters chosen for the BOBYQA and LBFGS optimisers. The performance of the SA algorithm, on the other hand, is less sensitive to the choice of initial parameters for these cases. Therefore, it is more robust and requires less preliminary manual sensitivity analysis. As discussed earlier, such a manual analysis is rather time consuming as it involves testing different set of parameters and observing the results manually. Furthermore, in three out of four cases, the SA outperformed the other optimisation algorithms in terms of reaching the minimum JS divergence.

It should be noted that the optimizers used in this paper are reaching an optimum solution by utilising an iterative process. Hence, as the number of iterations increases, the probability of achieving a better solution increases as well. However, this would increase the total CPU time required for the optimisation. In the case study presented, 5 DS parameters were tuned to simulate the elevation values of the ore/waste interface. The simulation of 20,758 grid nodes took, on average, 4 s to perform one run using 8 threads. Using 30,000 simulations in the optimisation process, therefore, takes roughly 33.3 h to complete. If the method is applied for a problem with larger grid size, CPU time may become an important concern. Therefore, the number of iterations should be adjusted depending on the nature and the complexity of the simulations. Based on our experience obtained from this study, even 100 function calls with the SA seem to be adequate to find a better set of parameters than the visual inspection method. This can also be seen in the convergence plots of the SA for 30,000 function calls (Figure 9). Taking the total reduction in 30,000 function calls as a reference (from 0.180 to 0.067), the graphs show that 75% of the reduction that is achieved actually takes place within 100 function calls. Whereas, 99% is achieved in 7000 function calls. Therefore, if feasible, the optimisation can be set to run around seven to eight thousand function calls. Otherwise, a hundred function calls would also yield satisfactory results.

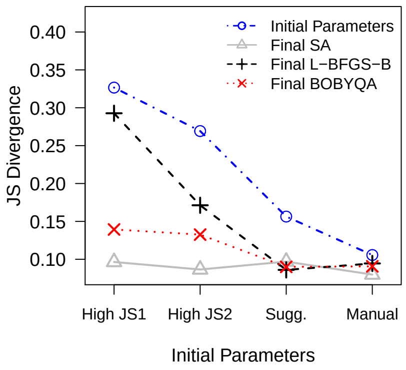

Figure 8. Convergence comparisons of three different optimisers with three different set of initial DS parameters.

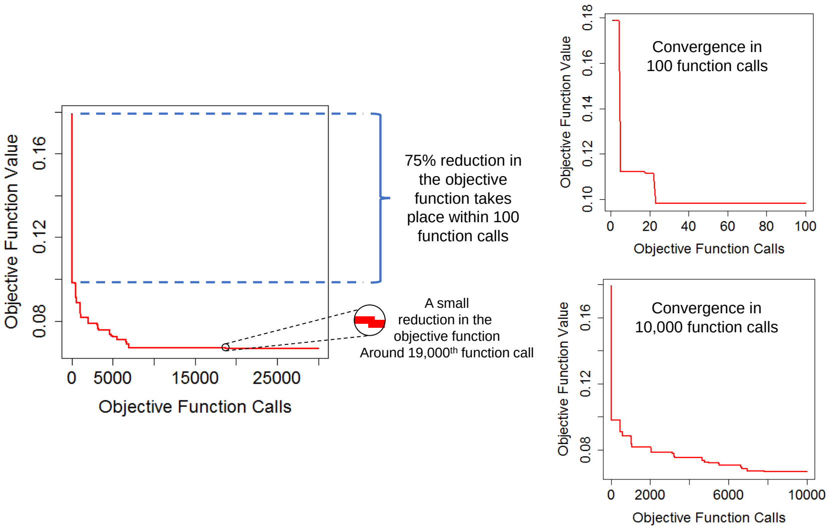

Figure 9. Convergence graphs obtained by running 30,000 function calls.

4.3. Effect of the Chosen Parameters on the Pattern Reproduction

Since the JS divergence is a function of five DS parameters, the direct visualisation of its behaviour depending on the parameters is not straightforward. An approach to visualise such multi-dimensional problems is to plot the lowest JS divergence values in 1-D against individual parameters. This technique is called profile-likelihood and is utilised to check visually the parameters of a statistical model obtained via maximum likelihood techniques. The idea is to plot the maximum likelihood function values as a function of all but one parameter. It can also be considered analogous to tracking the maximum values along the crests of a five dimensional function. Given a model with parameters (α, ψ) and the likelihood $L(\alpha, \psi)$, the profile of the likelihood $L_p(\alpha)$ for parameter α can be denoted as follows [43]:

$$L_p(\alpha) = L(\alpha, \hat{\psi}(\alpha)) = \max_{\psi}(L(\alpha, \psi(\alpha))) \tag{12}$$

To create such 1-D profiles, 30,000 DS simulations were first performed. During these simulations, the tested parameters and their associated JS divergence values were recorded. In our case, we were interested in minimising the objective function. Therefore, we used the created data set to plot the individual parameters against the minimum JS divergence values.

The created profiles shown in Figure 10 help our understanding of the optimisation problem better and its sensitivity to the parameters. For instance, they show that as the parameter t_{hd} increases, the JS divergence increases as well. This is an expected outcome as a high t_{hd} would lead to poor reproduction of the conditioning patterns; hence, would yield a high JS divergence.

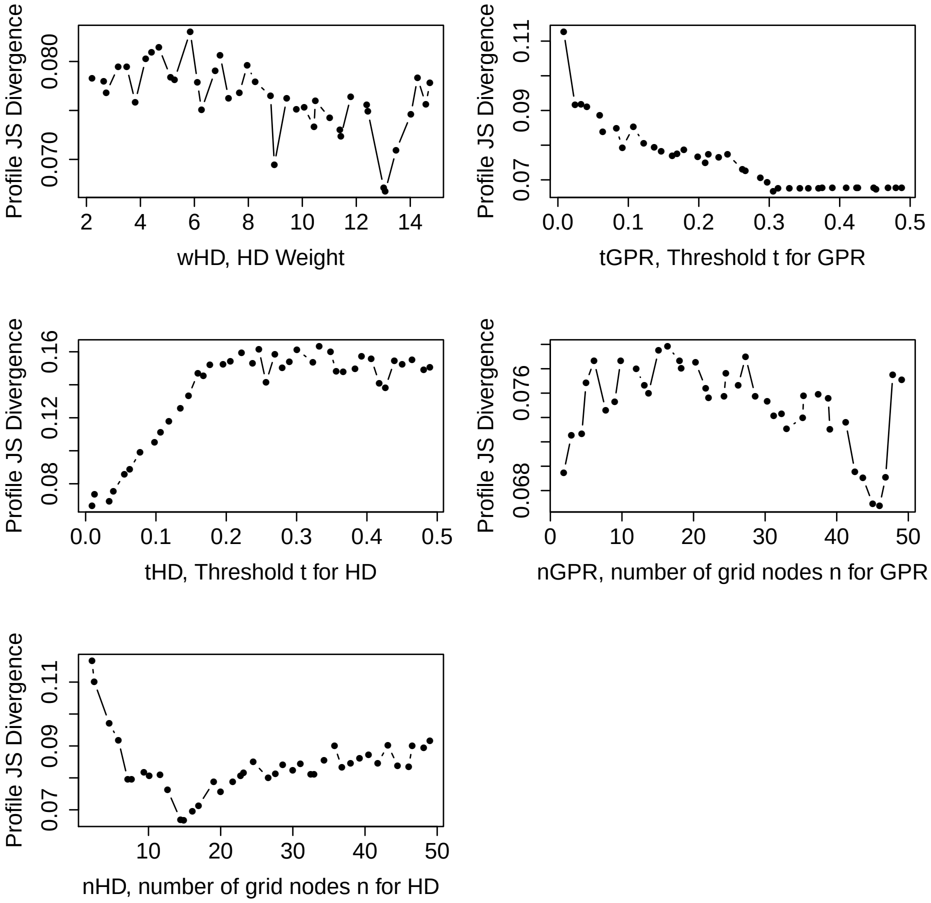

Figure 10. Profile JS divergences as a function of 5 DS input parameters.

For the n_{hd} parameter, the JS divergence decreases as expected with the maximum number of neighbours in the search template. But, the decrease stops at around thirteen and then starts to slightly increase with increasing n_{hd}. It means that the conditioning data patterns are best reproduced with search templates consisting of approximately thirteen informed nodes. This is unusual since in most cases the patterns of the TI are better reproduced with more points [20]. Here, the TI seems to contain features that are not present in the conditioning punctual data. One can see this as an indication that the TI is not fully compatible with the point data. But also, a very interesting feature is that the proposed automatic parameter identification is able to find automatically the best compromise between reproducing the patterns from the TI and those of the available point data. It is able to adapt the level of reproduction of the TI to ensure compatibility with the conditioning data.

Similarly, there is an apparent trend in the profile JS divergence for t_{gpr}. As the t_{gpr} increases, the JS divergence decreases and yields better simulation results. This is an unusual behaviour and can be better understood by being reminded of how the DS works. Considering a bivariate pattern collected from the SG, the grid nodes of the TI are visited to find a compatible match. During this search, a compatible pattern is considered to be the one with computed distances lower than the specified t_{hd} and t_{gpr}. An increase in t_{gpr} increases the probability of accepting a GPR component of the multivariate pattern as compatible. Therefore, the effect of the GPR variable becomes less pronounced in the simulations. Secondary information is normally considered to enhance the quality of the simulations. Here, the TI could be imperfectly compatible with the simulated area, as the multiple-point dependence between the reference mine and the simulation area could be different.

For the n_{GPR} and w_{HD} parameters, the plots are noisy and do not exhibit a simple systematic decrease or increase. Considering that these plots were created using 30,000 simulation data, these noisy plots should not be due to insufficient data. A more likely cause could be the existence of many local minima. This could explain why the SA performed more efficiently than the other optimization methods.

Another observation is related to the sensitivity of the JS divergence to the parameters. When the range of the JS divergence values are considered, t_{hd} seems to be the most sensitive parameter as it has a high range of JS divergence values. This parameter is followed by n_{hd} and t_{gpr}. The parameters w_{hd} and n_{gpr} seem to have less effect on the resulting JS divergences. Another interesting observation is related to the t_{hd} and t_{gpr} values. Both threshold values lose their significance on the JS divergence values above $t = 0.3$. Therefore, it can be concluded that the most influential range for these parameters for this case is between 0 and 0.3.

4.4. Analysis of the Automatically Tuned Parameters

The DS algorithm we used in this study was DeeSse. This version of the DS incorporates some modifications such as the introduction of a specific threshold for each variable rather than a global one as in the original version of Mariethoz et al. [10]. Therefore, a direct comparison of the parameters we found with the parameters suggested by Meerschman et al. [20] (based on the original DS) cannot be made. However, we can still get an insight into the parameters from their study. For instance, they have documented an increase in the pattern consistency by altering the conditioning data weight from 1 to 5. In our case, the optimal value was found to be 13.07. This means that high importance was given to the conditioning data points. Hence, better reproduction of the conditioning data pattern was attained. Also, the threshold value for a good quality simulation is suggested to be lower than 0.1 for the continuous variables. In addition, $t \geq 0.2$ is stated to produce noisy images. Our automatically found threshold values for the conditioning data, which is 0.009 is also consistent with their findings. However, GPR threshold t_{GPR} was determined as 0.305 and it is higher than their suggested values. The fact that t_{GPR} value is higher than t_{HD} could be due to the objective function chosen. As the aim was to reproduce the conditioning data better in the simulations, the focus has automatically become more on better pattern consistency of the reference mined-out surface patterns.

5. Summary and Conclusions

In this paper, we presented a methodology to improve bauxite footwall contact simulations through an automatic parameter tuning process. The main idea was to identify the parameters best reproducing the patterns existing in the borehole/conditioning data. There are several benefits of relying on the borehole data patterns to tune the parameters. First, although the borehole data is a partial representation of the ground truth, they serve as precise and rather reliable information. Second, in lateritic bauxite deposits, the data set can be dense enough to compute the pattern statistics of the conditioning data as a reference to be targeted in the simulations. Third, getting a TI may sometimes be difficult. Considering the mined-out topography exposed after a bauxite extraction, the resulting surface might not be entirely compatible with the conditioning data of another area. Therefore, putting more emphasis on the conditioning data and borrowing the patterns from a TI can be considered as a merge between two sets of information. As a result, the adapted MPS parameters lead to generating realisations which contain the correct patterns observed at the location where the forecast is being made.

The proposed approach mainly provides three significant advantages. The first advantage is related to the uncertainty estimation. The evolution of the calculated IQR values at each grid node reveals a fair drop in the uncertainty after the parameter optimisation. Also, the observed convergence of the simulation variograms towards the conditioning data variogram indicates that the uncertainty estimation is more reliable after optimisation. Secondly, the parameter selection is carried out more objectively than with the usual trial and error approach. Identification of the appropriate parameters is performed based on an objective and quantitative criterion. Therefore, it naturally alleviates any subjective decision that can be made as a result of the visual inspection method. Third, the approach reduces the time and effort spent for the sensitivity analysis. Contrary to a traditional parameter tuning procedure in which a set of different parameters are tested and the results are analysed manually, the proposed approach selects the parameters automatically. This reduces the time and effort spent for the manual sensitivity analysis. Only some initial tests are required to set up the optimisation process.

In addition to the above-mentioned advantages, the selection of the pseudo-histograms of Melnikova et al. [27] as pattern statistics provides: (1) the ability to calculate the pattern statistics of continuous images; (2) computation of the pattern statistics of regularly spaced borehole data and (3) an extensive flexibility when constructing the histogram categories. For instance, apart from the patterns of the borehole data, one can also include the patterns from the other sources to update the pattern categories. Examples of such sources may include the training images or simulations. All the patterns coming from different sources can then be used to construct the categories of the pseudo-histograms.

Visualisation of the behaviour of the JS divergence values based on the parameters has been performed through the profile-likelihood method. It allowed the analysis and better understanding of the behaviour of the objective function. The results have shown that the most influential parameters were t_{hd}, t_{gpr} and n_{hd}. An interesting observation was made on t_{gpr}. Contrary to the expectation, the JS divergence value increased with decreasing t_{gpr}. Such a behaviour could be due to a possible TI compatibility issue and should be further investigated. In general, we believe that the observed objective function behaviours are related to both the choice of objective function as well as the case study. For instance, if the chosen objective function included the better reproduction of the TI patterns rather than the conditioning data, a different result could have been observed. Therefore, a different case study and the choice of the objective function could lead to a different behaviour.

One of the drawbacks of the proposed approach is the requirement of regularly spaced conditioning data to calculate the pattern statistics. We use the patterns with a particular spatial configuration in the pseudo-histogram construction. Therefore, the proposed method can only be used if the conditioning data is regularly spaced. Considering the stratified deposits such as lateritic bauxite, nickel or coal seams, a regularly spaced exploration borehole configuration is often implemented. Hence, the proposed approach can be used to aid parameter tuning for the MPS algorithms for such deposits. However, the conditioning data sets may sometimes be sparse and it may not be possible to accurately infer the multiple-point statistics of the conditioning data and make a comparison with the simulations. In such a case, the objective function could focus more on the reproduction of the multiple-point statistics of the training image and can easily be integrated into the objective function in the manner described in this paper.

The general workflow proposed in the paper could be adapted to irregularly spaced conditioning data, but this would require using a different quantification of pattern statistics. One idea could be to use cumulants [8], because of the tolerances it offers for irregular grids. Another option would be to define a very different objective function focusing on a cross-validation approach where the quality criteria would be based on evaluating the MPS model performances. Most likely, the simulated annealing would still remain the most efficient algorithm to obtain the parameters since previous experiences have shown that the objective functions may contain local minima [20].

Lastly, as this approach identifies the parameters iteratively, it requires computational time. This is a well-known feature of the SA technique. However, the strength of the approach is that it allows obtaining a global minimum even when the objective function contains many local minima. This has also been observed by the comparison made using different optimisers with different initial DS parameters. The SA does not get stuck in local minimum values as compared to the other methods. Moreover, when different the optimisations are initiated with initial input parameters, the SA appears to be the least sensitive method among the ones tested. Therefore, it might require less preliminary sensitivity analysis to define the initial parameters for the optimisations. These facts favour the use of the SA to tune the parameters automatically.

Author Contributions: Y.D., P.R and J.S. designed the experiments and analysed the data; Y.D. performed the experiments and wrote the paper; O.E. and E.T. supervised the overall research study and edited the manuscript.

Acknowledgments: The authors would like to thank two anonymous reviewers for their comments and suggestions. The authors also kindly appreciate the financial support provided by the Office of Graduate Research School of Curtin University for publishing this paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SG	Simulation Grid
MPS	Multiple-Point Statistics
TI	Training Image
DS	Direct Sampling
SA	Simulated Annealing
JS	Jensen-Shannon Divergence
L-BFGS-B	Limited Memory Broyden-Fletcher-Goldfarb-Shanno Optimisation with Box Constraints
BOBYQA	Bound Optimisation by Quadratic Approximation

Glossary of Notations

$Z^*(x)$	Simulated value at location x
$d_n(x, L)$	Data event constructed at location x of the simulation grid
$d_n(y, L)$	Data event constructed at location y of the training image grid
$Z(y)$	Value of the attribute at y location of the training image
t	Acceptance threshold to accept a pattern compatible
f	Fraction of the training image to be scanned
$d\{d_n(x, L), d_n(y, L)\}$	Distance between the conditioning data and training image patterns
α_i	Weighting factor applied to the ith data event node
δ	Power for computing the α_i weighting factor
w	Special weight attached to the conditioning data events
T	Search template used to construct the pseudo-histograms
$pat_j^{TI,un}$	jth unique pattern collected from the training image
$H^{d,m}$	Histogram class of the model image
$H^{d,TI}$	Histogram class of the training image
pat_i^m	Pattern i observed in the model image
$N^{TI,un}$	Number of unique pattern categories in the training image
$d_{JS}(p, q)$	Jensen-shannon divergence of p and q density distributions
$f(x)$	Objective function
Δ	Change in the objective function due to a candidate solution x^{i+1}
pat_i^C	ith conditioning pattern collected from the simulation grid
$\overline{pat_i^C}$	Mean of the values at the nodes of the pattern pat_i^C
N^C	Number of patterns captured from the SG.
pat_i^{SIM}	Pattern i collected from the simulation grid
$\overline{pat_i^{SIM}}$	Mean value of the nodes in pattern pat_i^{SIM}
$H_j^{d,SIM}$	Value of the jth category of the realisation image pseudo-histogram
$H^{d,SIM}$	Pseudo-histogram of a realisation
N^{SIM}	Number of patterns captured in the realisation image generated
$H_j^{d_{Nor}}$	Normalised weight in the jth histogram category
$H_j^{d,C_{Nor}}$	Normalised weight of the jth category in the conditioning data pseudo-histogram
$H_j^{d,SIM_{Nor}}$	Normalised weight of the jth category in the simulation image pseudo-histogram
w_{hd}	Weighting factor attached to the hard data
w_{GPR}	Weighting factor attached to the GPR data
n_{hd}	Maximum number hard data nodes to be used in the MPS simulations
n_{GPR}	Maximum number GPR nodes to be used in the MPS simulations
t_{hd}	Acceptance threshold for the hard data
t_{GPR}	Acceptance threshold for the GPR data
$L(\alpha, \psi)$	Likelihood as a function of model parameters (α, ψ)
$L_p(\alpha)$	Profile of the likelihood as a function of the parameter α

References

1. Strebelle, S. Conditional simulation of complex geological structures using multiple-point statistics. *Math. Geol.* **2002**, *34*, 1–21. [CrossRef]
2. Oriani, F.; Straubhaar, J.; Renard, P.; Mariethoz, G. Simulation of rainfall time series from different climatic regions using the direct sampling technique. *Hydrol. Earth Syst. Sci.* **2014**, *18*, 3015. [CrossRef]
3. Pirot, G.; Straubhaar, J.; Renard, P. Simulation of braided river elevation model time series with multiple-point statistics. *Geomorphology* **2014**, *214*, 148–156. [CrossRef]
4. Guardiano, F.; Srivastava, R. *Borrowing Complex Geometries from Training Images: The Extended Normal Equations Algorithm*; Stanford Center for Reservoir Forecasting Report; Stanford University: Stanford, CA, USA, 1992.
5. Zhang, T.; Switzer, P.; Journel, A. Filter-based classification of training image patterns for spatial simulation. *Math. Geol.* **2006**, *38*, 63–80. [CrossRef]

6. Arpat, G.B.; Caers, J. Conditional simulation with patterns. *Math. Geol.* **2007**, *39*, 177–203. [CrossRef]
7. Gloaguen, E.; Dimitrakopoulos, R. Two-dimensional conditional simulations based on the wavelet decomposition of training images. *Math. Geosci.* **2009**, *41*, 679–701. [CrossRef]
8. Dimitrakopoulos, R.; Mustapha, H.; Gloaguen, E. High-order statistics of spatial random fields: Exploring spatial cumulants for modeling complex non-Gaussian and non-linear phenomena. *Math. Geosci.* **2010**, *42*, 65–99. [CrossRef]
9. Honarkhah, M.; Caers, J. Stochastic simulation of patterns using distance-based pattern modeling. *Math. Geosci.* **2010**, *42*, 487–517. [CrossRef]
10. Mariethoz, G.; Renard, P.; Straubhaar, J. The direct sampling method to perform multiple-point geostatistical simulations. *Water Resour. Res.* **2010**, *46*. [CrossRef]
11. Straubhaar, J.; Renard, P.; Mariethoz, G.; Froidevaux, R.; Besson, O. An improved parallel multiple-point algorithm using a list approach. *Math. Geosci.* **2011**, *43*, 305–328. [CrossRef]
12. Tahmasebi, P.; Hezarkhani, A.; Sahimi, M. Multiple-point geostatistical modeling based on the cross-correlation functions. *Comput. Geosci.* **2012**, *16*, 779–797. [CrossRef]
13. Erten, O. Profiling and Mining Control to Mitigate Dilution Effect from SiO_2 at the Base of a Bauxite Deposit. Ph.D. Thesis, The University of Queensland, Brisbane, Australia, 2012.
14. Dagasan, Y.; Erten, O.; Topal, E. Accounting for a spatial trend in fine-scale ground-penetrating radar data: A comparative case study. *J. S. Afr. Inst. Min. Metall.* **2018**, *118*, 173–184.
15. Dagasan, Y.; Renard, P.; Straubhaar, J.; Erten, O.; Topal, E. Pilot Point Optimization of Mining Boundaries for Lateritic Metal Deposits: Finding the Trade-off Between Dilution and Ore Loss. *Nat. Resour. Res.* **2018**, 1–19. [CrossRef]
16. Mariethoz, G.; Caers, J. *Multiple-Point Geostatistics: Stochastic Modeling with Training Images*; John Wiley & Sons: Hoboken, NJ, USA, 2014.
17. Boucher, A. Algorithm-Driven and Representation-Driven Random Function: A New Formalism for Applied Geostatistics, 2007. Avaliable online: https://pdfs.semanticscholar.org/0d38/a38d3b5c9ed694b788f9c3aab1fad381be7c.pdf (accessed on 21 May 2018).
18. Boisvert, J.B.; Pyrcz, M.J.; Deutsch, C.V. Multiple point metrics to assess categorical variable models. *Nat. Resour. Res.* **2010**, *19*, 165–175. [CrossRef]
19. Liu, Y. Using the Snesim program for multiple-point statistical simulation. *Comput. Geosci.* **2006**, *32*, 1544–1563. [CrossRef]
20. Meerschman, E.; Pirot, G.; Mariethoz, G.; Straubhaar, J.; Van Meirvenne, M.; Renard, P. A practical guide to performing multiple-point statistical simulations with the Direct Sampling algorithm. *Comput. Geosci.* **2013**, *52*, 307–324. [CrossRef]
21. Rezaee, H.; Mariethoz, G.; Koneshloo, M.; Asghari, O. Multiple-point geostatistical simulation using the bunch-pasting direct sampling method. *Comput. Geosci.* **2013**, *54*, 293–308. [CrossRef]
22. Huysmans, M.; Dassargues, A. Direct multiple-point geostatistical simulation of edge properties for modeling thin irregularly shaped surfaces. *Math. Geosci.* **2011**, *43*, 521. [CrossRef]
23. De Iaco, S.; Maggio, S. Validation techniques for geological patterns simulations based on variogram and multiple-point statistics. *Math. Geosci.* **2011**, *43*, 483–500. [CrossRef]
24. Tan, X.; Tahmasebi, P.; Caers, J. Comparing training-image based algorithms using an analysis of distance. *Math. Geosci.* **2014**, *46*, 149–169. [CrossRef]
25. Renard, P.; Allard, D. Connectivity metrics for subsurface flow and transport. *Adv. Water Resour.* **2013**, *51*, 168–196. [CrossRef]
26. Pérez, C.; Mariethoz, G.; Ortiz, J.M. Verifying the high-order consistency of training images with data for multiple-point geostatistics. *Comput. Geosci.* **2014**, *70*, 190–205. [CrossRef]
27. Melnikova, Y.; Zunino, A.; Lange, K.; Cordua, K.S.; Mosegaard, K. History matching through a smooth formulation of multiple-point statistics. *Math. Geosci.* **2015**, *47*, 397–416. [CrossRef]
28. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*; John Wiley & Sons: Hoboken, NJ, USA, 2012.
29. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [CrossRef] [PubMed]
30. Straubhaar, J. *DeeSse User's Guide*; The Centre for Hydrogeology and Geothermics (CHYN), University of Neuchâtel: Neuchâtel, Switzerland, 2016.

31. Straubhaar, J.; Renard, P.; Mariethoz, G. Conditioning multiple-point statistics simulations to block data. *Spat. Stat.* **2016**, *16*, 53–71. [CrossRef]
32. Endres, D.M.; Schindelin, J.E. A new metric for probability distributions. *IEEE Trans. Inf. Theory* **2003**, *49*, 1858–1860. [CrossRef]
33. Reeves, C.R. *Modern Heuristic Techniques for Combinatorial Problems*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1993.
34. Xiang, Y.; Gubian, S.; Martin, F. Generalized Simulated Annealing. In *Computational Optimization in Engineering-Paradigms and Applications*; InTech: London, UK, 2017.
35. Metropolis, N.; Rosenbluth, A.W.; Rosenbluth, M.N.; Teller, A.H.; Teller, E. Equation of state calculations by fast computing machines. *J. Chem. Phys.* **1953**, *21*, 1087–1092. [CrossRef]
36. Sun, N.Z.; Sun, A. *Model Calibration and Parameter Estimation: For Environmental and Water Resource Systems*; Springer: Berlin, Germany, 2015.
37. Szu, H.; Hartley, R. Fast simulated annealing. *Phys. Lett. A* **1987**, *122*, 157–162. [CrossRef]
38. Tsallis, C.; Stariolo, D.A. Generalized simulated annealing. *Phys. A Stat. Mech. Appl.* **1996**, *233*, 395–406. [CrossRef]
39. Team, R.C. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria. 2016, 2017.
40. Xiang, Y.; Gubian, S.; Suomela, B.; Hoeng, J. Generalized Simulated Annealing for Global Optimization: The GenSA Package. *R J.* **2013**, *5*, 13–28.
41. Zhu, C.; Byrd, R.H.; Lu, P.; Nocedal, J. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Softw. (TOMS)* **1997**, *23*, 550–560. [CrossRef]
42. Powell, M.J. *The BOBYQA Algorithm for Bound Constrained Optimization without Derivatives*; Cambridge NA Report NA2009/06; University of Cambridge: Cambridge, UK, 2009, pp. 26–46.
43. Diggle, P.J.; Tawn, J.; Moyeed, R. Model-based geostatistics. *J. R. Stat. Soc. Ser. C (Appl. Stat.)* **1998**, *47*, 299–350. [CrossRef]