A Method for Assessing Parameter Impact on Control-Flow Discovery Algorithms

Joel Ribeiro[1] and Josep Carmona[1]

Universitat Politècnica de Catalunya, Spain.
{jribeiro, jcarmona}@cs.upc.edu

Abstract. Given an event log L, a control-flow discovery algorithm f, and a quality metric m, this paper faces the following problem: what are the parameters in f that mostly influence its application in terms of m when applied to L? This paper proposes a method to solve this problem, based on *sensitivity analysis*, a theory which has been successfully applied in other areas. Clearly, a satisfactory solution to this problem will be crucial to bridge the gap between process discovery algorithms and final users. Additionally, recommendation techniques and meta-techniques like determining the *representational bias* of an algorithm may benefit from solutions to the problem considered in this paper. The method has been evaluated over a set of logs and the flexible heuristic miner, and the preliminary results witness the applicability of the general framework described in this paper.

1 Introduction

Control-flow discovery is considered as one of the crucial features of Process Mining [13]. Intuitively, discovering the control-flow of a process requires to analyze its executions and extract the causality relations between activities which, taken together, illustrate the structure and ordering of the process under consideration.

There are many factors that may hamper the applicability of a control-flow discovery algorithm. On the one hand, the log characteristics may induce the use of particular algorithms, e.g., in the presence of *noise* in the log it may be advisable to consider a noise-aware algorithm. On the other hand, the *representational bias* of an algorithm may hinder its applicability for elicitating the process underlying in a log.

Even in the ideal case where the more suitable control-flow discovery algorithm is used for tackling the discovery task, it may be the case that the default algorithm's parameters (designed to perform well over different scenarios) are not appropriate for the log at hand. In that case, the user is left alone in the task of configuring the best parameter values, a task which requires a knowledge of both the algorithm and the log at hand.

In this paper we present a method to automatically assess the impact of parameters of control-flow discovery algorithms. In our approach, we use an efficient technique from the discipline of sensitivity analysis for exploring the parameter search space. In the next section, we charaterize this sensitivity analysis technique and relate it with other work in the literature for similar purposes done in other areas.

We consider three direct applications of the method presented in this paper:

(A) As an aid to users of control-flow discovery algorithms: given a log, an algorithm and a particular quality metric the user is interested in, a method like the one presented in this paper will indicate the parameters to consider. Then the user will be able to influence (by assigning meaningful values to these parameters) the discovery experiment.

(B) As an aid for recommending control-flow discovery algorithms: current recommendation systems for control-flow process discovery (e.g., [9]) do not consider the parameters of the algorithms. Using the methodology of this paper, one may determine classes of parameters whose impact refer to the same quality metric, and those can be offered as modes of the same algorithm tailored to specific metrics. Hence, the recommendation task (i.e., the selection of a discovery algorithm) may then be guided towards a better use of a control-flow technique.

(C) As a new form of assessing the representational bias of an algorithm: given a log and an algorithm, it may well be the case that the impact of most of the algorithm's parameters is negligible. In that case, then if additionally the result obtained is not satisfactory, one may conclude that this is not the right algorithm for the log at hand.

The rest of the paper is organized as follows: Section 2 illustrates the contribution and provides related work. Section 3 provides the necessary background and main definitions. Then, Section 4 presents the main methodology of this paper, while Section 5 provides a general discussion on its complexity. Finally, Section 6 concludes the paper.

2 Related Work and Contribution

The selection of parameters for executing control-flow algorithms is usually a challenging issue. The uncertainty of the inputs, the lack of information about parameters, the diversity of outputs (i.e., the different process model types), and the difficulty of choosing a comprehensive quality measurement for assessing the output of a control-flow algorithm make the selection of parameters a difficult task.

The *parameter optimization* is one of the most effective approaches for parameter selection. In this approach, the parameter space is searched in order to find the best parameters setting with respect to a specific quality measure. Besides the aforementioned challenges, the main challenge of this approach is to select a robust strategy to search the parameter space. Grid (or exhaustive) search, random search [2], gradient descent based search [1] and evolutionary computation [7] are typical strategies, which have proven to be effective in optimization problems, but they are usually computationally costly. [16,6,3] are examples of parameter optimization applications on a control-flow algorithm. Besides the fact that only a single control-flow algorithm is considered, all of these approaches rely on quality measurements that are especially designed to work on a specific type of process model.

A different approach, which may also be used to facilitate the parameter optimization, is known as *sensibility analysis* [11] and consists of assessing the influence of the inputs of a mathematical model (or system) on the model's output. This information may help on understanding the relationship between the inputs and the output of the model, or identifying redundant inputs in specific contexts. Sensibility methods range from variance-based methods to screening techniques [11]. One of the advantages of screening is that it requires a relatively low number of evaluations when compared to other approaches. The *Elementary Effect* (EE) method [8,4,5] is a screening technique for sensibility analysis that can be applied to identify non-influential parameters of computationally costly algorithms. In this paper, the EE method is applied to assess the impact of the parameters of control-flow algorithms.

3 Preliminaries

This section contains the main definitions used in this paper.

3.1 Event Log and Process Model

Process data describe the execution of the different process events of a business process over time. An *event log* organizes process data as a set of process instances, where a process instance represents a sequence of events describing the execution of activities (or tasks).

Definition 1 (Event Log). *Let T be a set of events, T^* the set of all sequences (i.e., process instances) that are composed of zero or more events of T, and $\delta \in T^*$ a process instance. An event log L is a set of process instances, i.e., $L \in \mathcal{P}(T^*)$.*[1]

A *process model* is an activity-centric model that describes the business process in terms of activities and their dependency relations. Petri nets, Causal nets, BPMN, and EPCs are examples of notations for modeling these models. For an overview of process notations see [13]. A process model can be seen as an abstraction of how work is done in a specific business. A process model can be discovered from process data by applying some control-flow algorithm.

3.2 Control-Flow Algorithm

A control-flow algorithm is a process discovery technique that can be used for translating the process behavior described in an event log into a process model. These algorithms may be driven by different discovery strategies and provide different functionalities. Also, the execution of a control-flow algorithm may be constrained (controlled) by some parameters.

[1] $\mathcal{P}(X)$ denotes the powerset of some set X.

Definition 2 (Algorithm). *Let L be an event log, P a list of parameters, and R a process model. An (control-flow) algorithm A is defined as a function $f^A : (L, P) \rightarrow R$ that represents in R the process behavior described in L, and it is constrained by P. The execution of f^A is designated as a* ***discovery experiment****.*

3.3 Quality Measure

A *measure* can be defined as a measurement that evaluates the quality of the result of an (control-flow) algorithm. A measure can be categorized as follows [13].

Simplicity measure: quantifies the results of an algorithm (i.e., a process model mined from a specific event log) in terms of readability and comprehension. The number of elements in the model is an example of a simplicity measure.

Fitness measure: quantifies how much behavior described in the log complies with the behavior represented in the process model. The fitness is 100% if the model can describe every trace in the log.

Precision measure: quantifies how much behavior represented in the process model is described in the log. The precision is 100% if the log contains every possible trace represented in the model.

Generalization measure: quantifies the degree of abstraction beyond observed behavior, i.e., a general model will accept not only traces in the log, but some others that generalize these.

Definition 3 (Measure). *Let R be a process model and L an event log. A measure M is defined by*

- *a function $g^M : (R) \rightarrow \mathbb{R}$ that quantifies the quality of R, or*
- *a function $g^M : (R, L) \rightarrow \mathbb{R}$ that quantifies the quality of R according to L.*

The execution of g^M is designated as a ***conformance experiment****.*

3.4 Problem Definition

Given an event log L, a control-flow algorithm A constrained by the list of parameters $P = [p_1 = v_1, ..., p_k = v_k]$, and a quality measure M: *Assess the impact of each parameter $p \in P$ on the result of the execution of A over L, according to M.*

4 The Elementary Effect Method

The *Elementary Effect* (EE) method [8,4,5] is a technique for sensibility analysis that can be applied to identify non-influential parameters of control-flow algorithms, which usually are computationally costly for estimating other sensitivity analysis measures (e.g., variance-based measures). Rather than quantifying the exact importance of parameters, the EE method provides insight into the contribution of parameters to the results quality.

One of the most efficient EE methods is based on Sobol quasi-random numbers [12] and a radial OAT strategy [5].[2] The main idea is to analyze the parameter space by performing experiments and assessing the impact of changing parameters with respect to the results quality. A Sobol quasi-random generator is used to determine a uniformly distributed set of points in the parameter space. Radial OAT experiments [5] are executed over the generated points to measure the impact of the parameters. This information can be used either (i) to guide on the parameters setup by prioritizing the parameters to be tuned, or (ii) as a first step towards parameter optimization.

4.1 Radial OAT Experiments

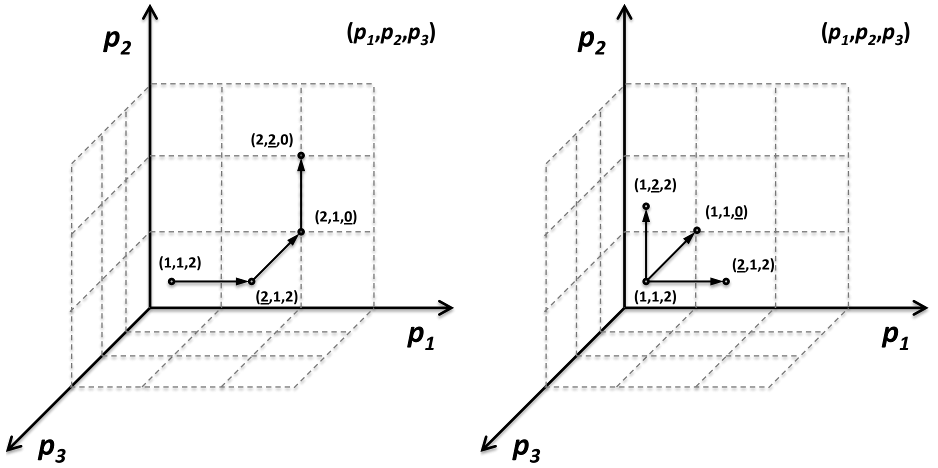
In this paper, an OAT experiment consists of a benchmark of some control-flow algorithm where the algorithm's parameters are assessed one at a time according to some quality measure. This means that $k + 1$ discovery and conformance experiments are conducted, the first to set a reference and the last k to compare the impact of changing one of the k algorithm's parameters. The parameter settings for establishing the reference and changing the parameter's values are defined by a pair of points from the parameter space. OAT experiments can use different strategies to explore these points. Figure 1 presents the most common strategies for performing OAT experiments. In the trajectory design, the parameter change compares to the point of the previous experiment. In the radial design, the parameter change compares always to the initial point. From these two, the radial design has been proven to outperform the trajectory one [10].

Radial OAT experiments can be defined as follows. First, a pair of points (α, β) is selected in the parameter space. Point α, the base point (point $(1, 1, 2)$ in Figure 1), is used as the reference parameter setting of the experiment. A discovery and conformance experiment is executed with this parameters setting to set the reference quality value. Point β, the auxiliary point (point $(2, 2, 0)$ in Figure 1), is used to compare the impact of changing the parameters, one at a time, from α to β. For each parameter $p_i \in P$, a discovery and conformance experiment is executed using the parameter values defined by α for a parameter $p_j \in P \wedge p_j \neq p_i$ and the parameter value defined by β for p_i (see the example in Figure 1b). Insight into the impact of each parameter is provided by aggregating the results of the radial OAT experiments.

Let A be a control-flow algorithm, M a given measure, and L an event log. The function $f^{A \cdot M}(L, P)$ computes the quality of the result of A over L with respect to M, where $P = [p_1 = v_1, ..., p_k = v_k]$ is the list of parameters of A.

$$f^{A \cdot M}(L, P) = \begin{cases} g^M(f^A(L, P)) & \text{if } M \text{ does not depend on a log} \\ g^M(f^A(L, P), L) & \text{otherwise} \end{cases} \tag{1}$$

[2] OAT stands for One (factor) At a Time.



(a) Trajectory. **(b)** Radial.

Fig. 1: Comparison between radial and trajectory samplings for OAT experiments over 3 parameters, using the points $(1, 1, 2)$ and $(2, 2, 0)$. The underlined values identify the parameter being assessed

The elementary effect of a parameter $p_i \in P$ on a radial OAT experiment is defined by

$$EE_i = \frac{f^{A \cdot M}(L, \alpha) - f^{A \cdot M}(L, \alpha \hookleftarrow \alpha_i \cdot \beta_i)}{\alpha_i - \beta_i}, \tag{2}$$

where α, β are parameter settings of P (the base and auxiliary points), α_i and β_i are the i^{th} elements of α and β, and $f^{A \cdot M}(L, \alpha \hookleftarrow \alpha_i \cdot \beta_i)$ is the function $f^{A \cdot M}(L, \alpha')$ where α' is α with β_i replacing α_i. The measure $\mu^\star$ for p_i is defined by

$$\mu_i^\star = \frac{\sum_{j=1}^{r} |EE_i|}{r}, \tag{3}$$

where r is the number of radial OAT experiments to be executed, typically between 10 and 50 [4]. The total number of discovery and conformance experiments is $r(k + 1)$, where k is the number of parameters of A.

The impact of a parameter $p_i \in P$ is given as the relative value of $\mu_i^\star$ compared to that for the other parameters of P. A parameter $p_j \in P$ $(j \neq i)$ is considered to have more impact on the results quality than p_i if $\mu_j^\star > \mu_i^\star$. The parameters p_j and p_i are considered to have equal impact on the results quality if $\mu_j^\star = \mu_i^\star$. The parameter p_i is considered to have no impact on the results quality if $\mu_i^\star = 0$. This measure is sufficient to provide a reliable ranking of the parameters [4,5].

4.2 Sobol Numbers

Sobol quasi-random numbers (or sequences) are low-discrepancy sequences that can be used to distribute uniformly a set of points over a multidimensional space. These sequences are defined by n points with m dimensions. Table 1 presents an example of a Sobol sequence containing ten points with ten dimensions.

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
x_1	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
x_2	0.7500	0.2500	0.2500	0.2500	0.7500	0.7500	0.2500	0.7500	0.7500	0.7500
x_3	0.2500	0.7500	0.7500	0.7500	0.2500	0.2500	0.7500	0.2500	0.2500	0.2500
x_4	0.3750	0.3750	0.6250	0.8750	0.3750	0.1250	0.3750	0.8750	0.8750	0.6250
x_5	0.8750	0.8750	0.1250	0.3750	0.8750	0.6250	0.8750	0.3750	0.3750	0.1250
x_6	0.6250	0.1250	0.8750	0.6250	0.6250	0.8750	0.1250	0.1250	0.1250	0.3750
x_7	0.1250	0.6250	0.3750	0.1250	0.1250	0.3750	0.6250	0.6250	0.6250	0.8750
x_8	0.1875	0.3125	0.9375	0.4375	0.5625	0.3125	0.4375	0.9375	0.9375	0.3125
x_9	0.6875	0.8125	0.4375	0.9375	0.0625	0.8125	0.9375	0.4375	0.4375	0.8125
x_{10}	0.9375	0.0625	0.6875	0.1875	0.3125	0.5625	0.1875	0.1875	0.1875	0.5625

Table 1: The first ten points of a ten-dimensional Sobol quasi-random sequence.

Each element of a point of a Sobol sequence consists of a numerical value between zero and one (e.g., the element representing the second dimension (d_2) of point x_5 is 0.8750). A collection of these values (the entire point or part of it) may be used to identify a specific point in a parameter space. An element of a point of a Sobol sequence can be converted into a parameter value by some normalization process. For instance, a possible normalization process for an element $e \in [0, 1]$ to one of the n distinct values of some discrete parameter p can be defined by $\lfloor e \times n \rfloor$, which identifies the index of the parameter value in p corresponding to e. Notice that the parameter space must be uniformly mapped by the normalization process (e.g., each value of a Boolean parameter must be represented by 50% of all possible elements).

Using the approach proposed in [5], a matrix of quasi-random Sobol numbers of dimensions $(r + 4{,}2k)$ can be used to analyze the elementary effects of the k parameters of a control-flow algorithm by executing r radial OAT experiments. The first k dimensions of the matrix's points define the base points, while the last k dimensions define the auxiliary points. Given that the first points of a Sobol sequence have the tendency to provide similar base and auxiliary points, it is identified in [5] the need of discarding the first four points of the sequence for the auxiliary points (i.e., the k rightmost columns should be shifted upward). Therefore, the base and auxiliary points can be computed from a Sobol sequence as follows. Let e_i^j be the element corresponding to the j^{th} dimension (d_j) of the i^{th} point (x_i) of the sequence. The i^{th} base (α^i) and auxiliary (β^i) points are defined as following.

$$\alpha^i = (e_i^1, e_i^2, ..., e_i^j) \ and \ \beta^i = (e_{i+4}^{j+1}, e_{i+4}^{j+2}, ..., e_{i+4}^{2j}). \tag{4}$$

4.3 Example: The FHM

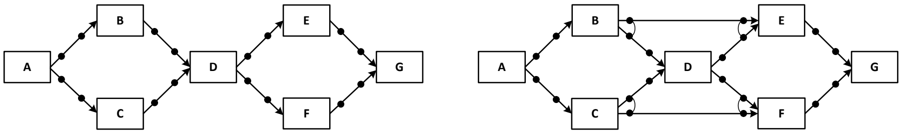
The following example is used to illustrate the analysis of the parameter space of an algorithm in order to assess the impact of the algorithm's parameters on the results quality. Let us consider an event log that is characterized by two distinct traces: $ABDEG$ and $ACDFG$. The frequency of any of these traces is high enough to not be considered as noise. The behavior described by these traces does not contain any kind of loop or parallelism, but it does contain two long-distance dependencies: $B \Rightarrow E$ and $C \Rightarrow F$. Let us also consider the Flexible Heuristics Miner (FHM) [17] as the control-flow algorithm to explore the parameter space in order to assess the impact of the FHM's parameters on the results quality. The parameters of the FHM are summarized in Table 2. Notice that every parameter of the FHM is continuous, with a range between zero and one. The *relative-to-best* and the *long-distance* thresholds are optional. The former is only considered with the *all-tasks-connected* heuristic. The latter is only taken into account when the *long-distance dependencies* option is activated.

Parameter	*Domain*	*Optional?*
Relative-to-best Threshold	$[0, 1]$	Yes
Dependency Threshold	$[0, 1]$	No
Length-one-loops Threshold	$[0, 1]$	No
Length-two-loops Threshold	$[0, 1]$	No
Long-distance Threshold	$[0, 1]$	Yes

Table 2: The parameters of the Flexible Heuristics Miner [17].

Figure 2 presents the two possible process models that can be mined with the FHM on the aforementioned event log, using all combinations of parameter values. Figure 2a shows the resulting Causal net where long-distance dependencies are not taken into account. Figure 2b shows the resulting Causal net with the long-distance dependencies. Notice that, depending on the quality measure, the quality of these process models may differ (e.g., the precision of the model with long-distance dependencies is higher than the other one). One may be interested on the exploration of the FHM's parameter space to get the process model that fulfills best some quality requirements.

The analysis of the parameter space of the FHM starts with the generation of the Sobol numbers. Let us consider that, for this analysis, one wants to execute $r = 30$ radial OAT experiments for assessing the elementary effects of the $k = 5$ FHM's parameters. So, a matrix of Sobol numbers of dimensions $(30 + 4{,}2 \times 5)$ has to be generated (cf. Section 4.2). Table 1 shows the first ten points of this matrix. Table 3 presents the first five base and auxiliary points as well as the parameter values corresponding to these points. Notice that the parameters are represented in the points according to the same ordering in Table 2 (i.e., the first element of a point represents the first parameter and so on). The normalization



(a) The Causal net without long-distance dependency relations.

(b) The Causal net with long-distance dependency relations.

Fig. 2: The process models that can be mined with the FHM.

process in this example is defined as follows. For the non-optional parameters (cf. Table 2), an element $e \in [0, 1]$ of a point of a Sobol sequence can be directly used to represent the value of the parameter. For the optional parameters, an element $e \in [0, 1]$ of a point of a Sobol sequence is normalized to a value $e' \in [0, 2]$, which maps the parameter space uniformly (i.e., the value of the parameter and whether or not the parameter is enabled). If $e' \leq 1$ then e' is assigned as the value of the parameter; the parameter is disabled otherwise.

Point	*Base*	*Auxiliary*
1	(.5000, .5000, .5000, .5000, .5000)	(.6250, .8750, .3750, .3750, .1250)
2	(.7500, .2500, .2500, .2500, .7500)	(.8750, .1250, .1250, .1250, .3750)
3	(.2500, .7500, .7500, .7500, .2500)	(.3750, .6250, .6250, .6250, .8750)
4	(.3750, .3750, .6250, .8750, .3750)	(.3125, .4375, .9375, .9375, .3125)
5	(.8750, .8750, .1250, .3750, .8750)	(.8125, .9375, .4375, .4375, .8125)
...	...	...

(a) The first five base and auxiliary points.

Point	*Base*	*Auxiliary*
1	(–, 0.50, 0.50, 0.50, –)	(–, 0.88, 0.38, 0.38, 0.25)
2	(–, 0.25, 0.25, 0.25, –)	(–, 0.13, 0.13, 0.13, 0.75)
3	(0.50, 0.75, 0.75, 0.75, 0.50)	(0.75, 0.63, 0.63, 0.63, –)
4	(0.75, 0.38, 0.63, 0.88, 0.75)	(0.63, 0.44, 0.94, 0.94, 0.63)
5	(–, 0.88, 0.13, 0.38, –)	(–, 0.94, 0.44, 0.44, –)
...	...	...

(b) The parameter values for the first five base and auxiliary points. The wildcard value '–' identifies that the parameter is disabled.

Table 3: The first five points of the Sobol numbers.

Table 4 presents the radial sampling for the first radial OAT experiment (first point in Table 3) as well as the result of the execution of $f^{A \cdot M}(L, P)$ and the elementary effect EE for each parameter. For executing $f^{A \cdot M}(L, P)$, A is the

FHM, M the *Node Arc Degree* measure[3], and L the aforementioned event log. The elementary effects are computed as described in Section 4.1.[4] Notice that the elementary effect of a parameter can only be computed when the base and auxiliary points provide distinct parameter values (e.g., in Table 4, the first parameter is not assessed because it is disabled in both base and auxiliary points).

Parameter Values	*Result*	*Elementary Effect*
P	$f^{A \cdot M}(L, P)$	EE_i
(–, 0.50, 0.50, 0.50, –)	2.154	
($\underline{-}$, 0.50, 0.50, 0.50, –)		
(–, $\underline{0.88}$, 0.50, 0.50, –)	2.154	0.0
(–, 0.50, $\underline{0.38}$, 0.50, –)	2.154	0.0
(–, 0.50, 0.50, $\underline{0.38}$, –)	2.154	0.0
(–, 0.50, 0.50, 0.50, $\underline{0.25}$)	2.316	0.162

Table 4: Radial sampling for the first radial OAT experiment. The first line corresponds to the base point, while the others consist of the base point in which the element regarding a specific parameter is replaced by that from the auxiliary point; the underlined values identify the replaced element and the parameter being assessed.

Table 5 presents the results of the analysis of the FHM's parameter space. The results identify the long-distance threshold as the only parameter to take into account for the parameter exploration. As expected, all other parameters have no impact on the results quality. This is explained by the fact that the log does not contain any kind of loop or noise. Notice that the $\mu^\star$ absolute value does not provide any insight into how much a parameter influences the results quality. Instead, the $\mu^\star$ measurement provides insight into the impact of a parameter on the results quality, compared to others.

Parameter	$\mu^\star$
Dependency Threshold	0.0
Relative-to-best Threshold	0.0
Length-one-loops Threshold	0.0
Length-two-loops Threshold	0.0
Long-distance Threshold	0.113

Table 5: The $\mu^\star$ values of the FHM's parameters.

[3] The *Node Arc Degree* measure consists of the average of incoming and outgoing arcs of every node of the process model.

[4] For computing EE_i, $\alpha_i - \beta_i$ is considered to be 1 when the parameter is changed from a disabled to an enabled state, or the other way around (e.g., the last parameter in Table 4).

5 Application

The EE method presented in the previous section can be applied to any control-flow algorithm constrained by many parameters, using some event log and a measure capable of quantifying the quality of the result of the algorithm. The presented method can be easily implemented on some framework capable of executing discovery and conformance experiments (e.g., ProM [15] or CoBeFra [14]). Several open-source generators of Sobol numbers are available on the web.

The computational cost of our approach can be defined as follows. Let L be an event log, A a control-flow algorithm constrained by the list of parameters $P = [p_1 = v_1, ..., p_k = v_k]$, and M a quality measure. The computational cost of a discovery experiment using A (with some parameter setting) over L is given by C_D. Considering R as the result of a discovery experiment, the computational cost of a conformance experiment over R and L (or just R) with regard to M is given by C_C. Therefore, the computational cost of a radial OAT experiment is given by $C_E = (k+1)(C_D + C_C)$, where k is the number of parameters of A. The computational cost of the EE method based on r radial OAT experiments is given by $C = r(k+1)(C_D + C_C)$.

5.1 Perfomance Optimization

Considering that both discovery and conformance experiments may be computationally costly, performance may become a critical issue for the application of this method. This issue can be partially addressed by identifying a set of potentially irrelevant parameters, and considering those parameters as a group. Then, by adjusting the $\mu^\star$ measurement to work with groups of two or more parameters [4], the group of parameters can be analyzed together using radial experiments that iterate over all elements of the same group simultaneously.

Suppose, for instance, that it is known that a given log does not have loops. So, for the FHM's parameters, the *length-one-loops* and *length-two-loops* thresholds may be grouped in order to avoid the execution of discovery and conformance experiments that are not relevant for the analysis. Recalling the example presented in Section 4.3, the radial experiments will iterate over one group of two parameters and three indepedent parameters (i.e., the *dependency*, the *relative-to-best*, and the *long-distance* thresholds). This means that, for the group of parameters, all elements of the same group are replaced simultaneously by the corresponding elements from the auxiliary point. Table 6 presents the adjusted radial sampling presented in Table 4. The first line corresponds to the base point, while the others consist of the base point in which the element(s) regarding a specific parameter (or group of parameters) is replaced by that from the auxiliary point; the underlined values identify the replaced element(s) and the parameter (or group of parameters) being assessed.

Parameter Values
(–, 0.50, 0.50, 0.50, –)
($\underline{–}$, 0.50, 0.50, 0.50, –)
(–, $\underline{0.88}$, 0.50, 0.50, –)
(–, 0.50, $\underline{0.38}$, $\underline{0.38}$, –)
(–, 0.50, 0.50, 0.50, $\underline{0.25}$)

Table 6: Radial sampling for the first radial experiment considering a group of parameters.

The elementary effect of a group of parameters $G \subseteq P$ on a radial experiment is defined by

$$EE_G = \frac{f^{A \cdot M}(L, \alpha) - f^{A \cdot M}(L, \alpha \hookleftarrow \alpha_G \cdot \beta_G)}{dist(\alpha_G, \beta_G)}, \tag{5}$$

where α, β are parameter settings of P (the base and auxiliary points), α_G and β_G are the elements of G in α and β, and $f^{A \cdot M}(L, \alpha \hookleftarrow \alpha_G \cdot \beta_G)$ is the function $f^{A \cdot M}(L, \alpha')$ where α' is α with β_G replacing α_G. The function $dist(A, B)$ computes the distance between A and B (e.g., the Euclidean distance). The measure $\mu^\star$ for G is defined by

$$\mu^\star_G = \frac{\sum_{j=1}^{r} |EE_G|}{r}, \tag{6}$$

where r is the number of radial experiments to be executed. The total number of discovery and conformance experiments depends on the number of groups and independent parameters being assessed.

6 Conclusions and Future Work

To the best of our knowledge, this work is the first in presenting a methodology to assess the impact of parameters in control-flow discovery algorithms. The method relies on a modern sensitivity analysis technique that requires considerably less exploration than traditional ones such as genetic algorithms or variance-based methods.

In this work, we have applied the methodology on the Flexible Heuristics Miner algorithm using 13 event logs. The results suggest the effectiveness of the method. We have noticed that simple conformance measures (and, thus, less computationally costly) are as good as any other complex measure for assessing the parameters influence. Nevertheless, we acknowledge that more experiments are necessary to get a better insight.

Future work is mainly oriented towards addressing three aspects, which are mainly addressed to apply the method of this paper to other control-flow algorithms. First, we are interested in the algorithmic perspective in order to study the most efficient form of assessing the impact of a parameter, with the method presented in this paper as a baseline. Second, we will try to incorporate the methodology described in this paper in the RS4PD, a recommender system for process discovery [9]. Finally, the application of the presented method with other goals, e.g., estimating the representational bias of control-flow discovery algorithms may be explored.

Acknowledgments. This work as been partially supported by funds from the Spanish Ministry for Economy and Competitiveness (MINECO) and the European Union (FEDER funds) under grant COMMAS (ref. TIN2013-46181-C2-1-R).

References

1. Y. Bengio. Gradient-Based Optimization of Hyperparameters. *Neural computation*, 12(8):1889–1900, 2000.
2. J. Bergstra and Y. Bengio. Random Search for Hyper-Parameter Optimization. *The Journal of Machine Learning Research*, 13(1):281–305, 2012.
3. A. Burattin and A. Sperduti. Automatic Determination of Parameters' Values for Heuristics Miner++. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8, July 2010.
4. F. Campolongo, J. Cariboni, and A. Saltelli. An Effective Screening Design for Sensitivity Analysis of Large Models. *Environmental Modelling & Software*, 22(10):1509 – 1518, 2007.
5. F. Campolongo, A. Saltelli, and J. Cariboni. From Screening to Quantitative Sensitivity Analysis. A Unified Approach. *Computer Physics Communications*, 182(4):978–988, 2011.
6. L. Ma. How to Evaluate the Performance of Process Discovery Algorithms: A Benchmark Experiment to Assess the Performance of Flexible Heuristics Miner. Master's thesis, Eindhoven University of Technology, Eindhoven, 2012.
7. Z. Michalewicz and M. Schoenauer. Evolutionary Algorithms for Constrained Parameter Optimization Problems. *Evolutionary Computation*, 4(1):1–32, March 1996.
8. M.D. Morris. Factorial Sampling Plans for Preliminary Computational Experiments. *Technometrics*, 33(2):161–174, April 1991.
9. J. Ribeiro, J. Carmona, M. Misir, and M. Sebag. A Recommender System for Process Discovery. In S. Sadiq, P. Soffer, and H. Vlzer, editors, *Business Process Management*, volume 8659 of *Lecture Notes in Computer Science*, pages 67–83. Springer International Publishing, 2014.
10. A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, and S. Tarantola. Variance Based Sensitivity Analysis of Model Output. Design and Estimator for the Total Sensitivity Index. *Computer Physics Communications*, 181(2):259 – 270, 2010.
11. A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global Sensitivity Analysis: The Primer.* Wiley, 2008.
12. I.M. Sobol. Uniformly Distributed Sequences With an Additional Uniform Property. *USSR Computational Mathematics and Mathematical Physics*, 16(5):236 – 242, 1976.
13. W.M.P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes.* Springer, Berlin, 2011.

14. S. vanden Broucke, J.D. Weerdt, B. Baesens, and J. Vanthienen. A Comprehensive Benchmarking Framework (CoBeFra) for conformance analysis between procedural process models and event logs in ProM. In *IEEE Symposium on Computational Intelligence and Data Mining*, Grand Copthorne Hotel, Singapore, 2013. IEEE.
15. H.M.W. Verbeek, J.C.A.M. Buijs, B.F. van Dongen, and W.M.P. van der Aalst. ProM 6: The Process Mining Toolkit. In *Demo at the 8th International Conference on Business Process Management*, volume 615 of *CEUR-WS*, pages 34–39. 2010.
16. A.J.M.M. Weijters. An Optimization Framework for Process Discovery Algorithms. In *Proceedings of the International Conference on Data Mining, Las Vegas, Nevada, USA*, 2011.
17. A.J.M.M. Weijters and J.T.S. Ribeiro. Flexible Heuristics Miner (FHM). In *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2011, Paris, France*. IEEE, 2011.