Lei Chen

Editor-in-Chief

IEEE Transactions on Knowledge and Data Engineering

31 mag 2022

Dear Editor-in-Chief, Associate Editor, and Reviewers,

thank you very much for the very prompt handling of our submission. The reviewers have been very constructive in pointing out several minor issues, as well as few important major issues.

We have done our best to address all of them. In particular, we have modified the Introduction to better clarify the framework and the aims of our work; we have added further details about the implementation (3.3) and a dedicated section for algorithm complexity considerations (3.4). Furthermore, we have enriched the related work (5) as suggested, and significantly extended the evaluation with other real-life logs.

We did our best to limit the length of the paper by avoiding multiple repetitions of the same concepts, but the numerous new contents introduced to answer the reviewer’s concerns forced us to increase the number of pages by 3. So, in case the paper is accepted, we kindly ask the Editor-in-Chief’s approval despite the current length of the paper, which is 17 pages.

A detailed account of all the modifications, and how the paper has been improved following the reviewer’s suggestions, is summarized in the following.

Our best regards,

The authors

Referee #1

*Recommendation: Author Should Prepare A Minor Revision*

*Comments:*

*1- The authors propose a method (Algorithm 1, Algorithm 2) without giving its time complexity using Big-O notation.*

*What is the time complexity of the method?*

*O(n2), O(nlogn) etc.*

First of all, we thank the reviewer for the precious comments. Regarding the complexity of the method, we added a specific discussion subsection (3.4 Complexity considerations) to clarify the complexity of both the algorithms.

*2- There are many symbols. To increase the readability of the paper, a notation table can be added to give all symbols and their meanings.*

We added a table with all the symbols (Table 1), hoping this increases the readability of the paper.

*3- This study is about "Process Mining". However, some important studies about this topic do not been discussed in the paper. In order to make a strong "Related Work" section, I suggest the authors citing the following paper related to "process mining" ("Interactive process miner: a new approach for process mining", 2018).*

We thank the reviewer for the suggested related work. We cited it in section 5 (citation number [46]) for its accurate systematic review of the state of the art of procedural approaches.

**Well known examples of procedural process discoverers are the ones presented in the works [7], [8], [9], [10], [42], [43], [44]. See [45], [46] for systematic literature reviews of this field.**

*Furthermore, providing a table that summarizes the related work would increase the understandability of the difference from the previous studies in the related works section.*

We followed the reviewer’s suggestion and added a table (Table 7) to help the reader visualizing the classification of the cited related works.

*4- Some abbreviations are used in the text without giving their expansion.*

*For example; SCIFF, SAT, AGNEs, CERV, SEPSIS, BPIC12, etc.*

*The authors should write that "these abbreviations stand for what".*

We expanded the reported abbreviations, except for CERV, SEPSIS, BPIC12 because these are the actual names of the available logs. For example, SEPSIS is an event log describing the paths of patients (probably) affected by sepsis in a Dutch hospital. However, in section 4.2 we reported the explanation of the log’s contents, which should give the reader an idea of the origin of the three names.

**BPIC12 is a real-life logs from the Business Process Intelligence Challenge 2012 [33]. The log pertains to the application process for personal loans or overdrafts in a Dutch financial institute. […]**

**CERV is an event log related to the process of cervical cancer screening carried out in an Italian cervical cancer screening centre. [...]**

**SEPSIS [37] is an event log that records trajectories of patients with symptoms of the life-threatening sepsis condition in a Dutch hospital. [...]**

The same was done for the other three datasets that we added following reviewer#3’s suggestion.

**The DREYERS log [36] is an event log documenting the application grant process of the Dreyers Foundation, a Danish foundation supporting budding lawyers and architects. […]**

**The production event log (PROD) [34] contains data from a manufacturing process. Each trace records information about activities, workers and machines involved in the production of items. […]**

**Finally, the TRAFFIC fines event log [38], which comes from an Italian local police force, contains events on fine notifications, as well as (partial) repayments.**

*5- The organization of the paper (the structure of the manuscript) may be written at the end of the "Introduction" section.*

*For example: "Section 2 presents ... Section 3 gives ...."*

Following the reviewer suggestion, we added a description of the structure of the paper towards the end of the introduction. We report the paragraph here for clarity.

**To explain our approach, we start by providing an overview of the background and relevant concepts in Section 2, then we describe the approach in detail in Section 3, and we evaluate it in Section 4. Related work and conclusion follow.**

Please note that, to improve the readability of the work, we also expanded the contribution paragraphs and put it into a separated section (1.1) as suggested also by reviewer#3.

*Additional Questions:*

*1. Please explain how this manuscript advances this field of research and/or contributes something new to the literature.: It presents a novel discovery approach, NegDis, based on the underlying logic semantics of Declare, which makes use of the information brought by the positive and negative*

*example sets to produce declarative models.*

*2. Is the manuscript technically sound? Please explain your answer under Public Comments below.: Yes*

*1. Which category describes this manuscript?: Research/Technology*

*2. How relevant is this manuscript to the readers of this periodical? Please explain your rating under Public Comments below.: Very Relevant*

*1. Are the title, abstract, and keywords appropriate? Please explain under Public Comments below.: Yes*

*2. Does the manuscript contain sufficient and appropriate references? Please explain under Public Comments below.: References are sufficient and appropriate*

*3. Does the introduction state the objectives of the manuscript in terms that encourage the reader to read on? Please explain your answer under Public Comments below.: Yes*

*4. How would you rate the organization of the manuscript? Is it focused? Is the length appropriate for the topic? Please explain under Public Comments below.: Satisfactory*

*5. Please rate the readability of the manuscript. Explain your rating under Public Comments below.: Readable - but requires some effort to understand*

*6. Should the supplemental material be included? (Click on the Supplementary Files icon to view files): Does not apply, no supplementary files included*

*7. If yes to 6, should it be accepted: After revisions. Please include explanation under Public Comments below.*

*Please rate the manuscript. Please explain your answer.: Excellent*

Referee #2

*Recommendation: Author Should Prepare A Major Revision For A Second Review*

*Comments:*

*While there is no question that this is an interesting research topic, there are two elements which makes this paper a borderline case.*

*Firstly, it seems to ignore the literature on supervised sequence mining as well as recent efforts in binary process discovery. In general, there exists quite an extensive literature on mining minimal sequences (e.g. [1,2]) towards the general exercise of (multiclass) classification. A dedicated approach for Declare constraints exists as well [3]. The proposed approach should at least be pitched against [1,2] in terms of scalability and expressiveness (the benefit of Declare constraints), however, many of the models shown are very simple and can be covered by partial orders. Then, the efficiency and number of constraints can be pitched against [3].*

*While it is true that the strength of the approach is that it can mine models (although the introduction states ‘set of constraints’), there is not guarantee that the approach actually delivers models where there is any interaction between the constraints that delivers more information than [1-3].*

We thank the reviewer for his careful reading of the manuscript and constructive remarks. He is certainly right that we did not position the paper correctly w.r.t. the state of the art. To address this problem, we have added a remark in the introduction to highlight the connections between our work and the field of sequence classification, and to better clarify our goal.

**Besides the declarative-procedural classification, process discovery approaches can be also divided into two categories according to their vision on the model-extraction task. As also pointed out by Ponce-de-Leòn et al. [6], the vast majority of works in the process discovery spectrum (e.g. [7], [8], [9], [10]) can be seen as a one-class supervised learning technique, while fewer works (e.g. [11], [12], [13], [14]) intend model extraction as a two-class supervised task—which is driven by the possibility of partitioning the log traces into two sets according to some business or domain-related criterion. In a sense, two-class process discovery can be related to sequence classification, which deals with the task of discriminating between classes of sequences by deriving sequential patterns from a temporal database. The main enhancement of process discovery is that it focuses on extracting behavioural constraints, which are in general much more informative (and concise), than a set of sequences [15]. Another significant difference is in the semantics of the classes, which for process discovery are usually referred to as *positive* and *negative* examples [6], [14], intending them as disjuncts sets of desirable and unwanted behaviours, respectively. In sequence classification instead, classes can have any meaning (and in general be more than just two, not necessarily disjunct). This also entails the slightly different goal of binary process discovery, that is to learn a model characterising the positive set while taking into account also the negative one.**

Furthermore, we pitched our approach against [1,2,3] in the related work section.

**Two-class declarative process discovery is sometimes related to sequence mining and classification [15], [63], [64], [65], [66]. In [63] the authors propose a probabilistic machine learning approach that infers subsequences which best compress a sequence database. A similar Bayesian technique is used by Egho et al. [64] to mine Standard Classification Rule Models, where the antecedent is the relevant subsequence, and the consequent expresses the support of that subsequence in each class. The above-mentioned work by Maggi et al. [47] originated from the idea that Apriori-like approaches can discover local patterns in a log, but not rules representing prohibited behaviours and choices. The same motivation is behind the work by De Smedt et al. [15], which significantly enriches the expressiveness of the discovered models employing Declare behavioural constraints in sequence classification. Differently from [15], our work does not focus on characterising the traces of multiple, possibly intersected, input classes, but rather on refining the relevant feature of the traces belonging to a single class (the positive). In order to do so, we assume two disjunct sets of examples and suggest extracting from one class (the negative) the information relevant to characterise the other (the positive). In other words, the information that is relevant for our purpose is the fact that a trace *does not belong* to a certain class. Another important difference regards the mining algorithm: while [15] explicitly considers each relevant Declare constraint, we envisage the language bias, as well as the subsumption hierarchy as input and we exploit the power of a satisfiability-based solver to deter- mine the best model in terms of generality or simplicity.**

The reviewer is also correct when pointing out that many of the models shown throughout the paper can be covered with partial order. After a discussion between the authors, we agreed not to change them, because they serve just as examples, for the reader to understand the procedure’s steps.

As a side note, we wish to clarify that—following the example of other works on declarative process mining—we intend a Declare model as a logical conjunction (AND formula) of Declare constraints. In this sense, the model is often seen as a set of constraints, and a trace must fulfil all of them (as if they were logically connected with ANDs) in order to be deemed compliant. This is the reason why in the introduction we said, “we hereby focus on learning a set of constraints that is able to reconstruct …”. Nonetheless we understand that the sentence was a little misleading at that point of the paper, so we slightly changed the paragraph as follows.

**For this reason, we hereby focus on learning a model that is able to reconstruct which traces belong to which set (by accepting all the positive and rejecting, if possible, all the negative), while reflecting the user expectations on the quality of the extracted model according to predefined metrics.**

*Indeed, the ‘models’ in Table 1 are just 1 constraint-models. This seems like a very trivial result and needing 211s seems a lot for finding that this 1 constraint perfectly discriminates between the models.*

The reviewer’s objection is right and pushed us to check again our implementation. Thus, we applied some simple changes that provided a significant improvement in the computation time. In the specific case, it is now 22.93 seconds.

Nonetheless, we agree it may still appear a considerable time w.r.t. to the task performed: we probably failed to clarify the origin of it. Indeed, we do not want to find a model which just admits all traces in L+ and rejects all in L-, but we want the best according to generality or simplicity preferences. Therefore, the algorithm firstly generates all models admitting all traces in L+ and rejecting all in L- (actually, all those that are possible to reject), and then performs an optimization. In the paper, we also added a brief comment to Table 2 (previously Table 1).

**Table 2 summarize the obtained results and reports the best selected model for each scenario. The Time column highlights how the most expensive task is the generation of the set. This is indeed expected, as it requires to check all constraints in against all traces in .**

Finally, another thing that maybe was misleading was the column’s header saying, “First Discovered Model”, while it should have been “Best Discovered Model”, so we changed that in the table.

*Besides, is the procedure of generating models guaranteed to produce models with non-conflicting constraints to obtain C and Z?*

Yes, it is. But the reviewer’s comment persuaded us that a further clarification is needed in this regard. So, we added the following sentence after Definition 3.2

**Note that condition *(ii*)[ i.e., we have ]in the definition above ensures that any candidate solution is consistent (i.e., there are no conflicting constraints) when is non-empty.**

And we remarked the concept in subsection 3.2

**By construction, any subset of is non-conflicting, because any trace in is accepted by all its constraints. This simplifies the optimisation step, since there is no need to verify whether any of the selected models are conflicting (see point *(ii)* of Definition 3.2).**

*Then, it also remains to be seen how the approach relates to the recent work in [4], which provides a general algorithm for binary mining. It even proves that for Declare a perfect binary miner is not achievable (theorem 10). Note that positive/negative cases are defined differently. It is only in the experimental evaluation that examples are provided concerning positive/negative cases (i.e., compliance and exceeding mean/median duration)*

Thanks for this reference too. We added a paragraph to discuss the ideas of [4] in the related work section.

**Another recent related work is the Rejection Miner by Slaats et al. [14], which—analogously to our vision—considers process discovery as a binary classification task and provides an algorithm where some parts can be customised to include more constraints and different model minimisation strategies. Likewise, our approach envisages the possibility to employ a different language bias or closure operator. Since the aim of [14] is to provide a general miner (not limited to Declare notation), they do not focus on notation-specific improvements like our use of the closure to avoid redundancies and provide more compact models. Interestingly, [14] also reports a demonstration of the reasons why Declare is not able to perfectly separate any pair of positive/negative traces. Consistently with this finding, our approach aims at discovering a model that accepts all traces in L+ and rejects as many traces in L− as possible. Despite its limits, we believe that Declare’s diffusion in both industry and academia motivates the need to further investigate process discovery with this language.**

*Secondly, the approach is relatively straightforward but many of the guarantees and operations required to optimally execute it are underlit. Specifically, it would be welcome to know how 6a/b-8a/b are calculated. They seem especially intractable when there are models with many constraints. The optimization cost to achieve them is included and seems relatively low, but how would this work for large models?*

The reviewer is right: we did not provide enough details regarding the implementation. To correct this, we expanded section 3.3 Implementation with the following paragraphs.

**The second optimisation stage has been implemented using the Answer Set Programming (ASP) system CLINGO [24]. The main reason for selecting an ASP system for finite domain optimisation is that rules provide an effective and intuitive framework to implement a large class of closure operators. Indeed, all the deductive systems for Declare that we analysed in the literature (see e.g. [22], [25]) are expressed in the form of logic rules c1∧ . . . ∧ cn =⇒ cn+1, where ci are Declare constraints or their negation; therefore, they can be equivalently expressed as Normal Logic Programs (NLPs) [26] by exploiting the assumption that the set of activities is finite and known in advance. The declarative semantics of NLPs ensures that the (unique) deductive closure of the rules is taken into account for the optimisation stage. For example the valid formula INIT(a) ∧ b!=a ⇒ PRECEDENCE(a,b) that holds for any pair of activities a, b can be written as the rule**

**precedence(A,B) :− init (A) , activity (B) , A != B.**

**using a specific predicate (activity/1) representing the set of all activities. \footnote{** **By leveraging CLINGO function symbols, Declare constraints are encoded as function terms; for example, the constraints derived are represented using the unary predicate holds/1: holds(precedence(a1,a2)) }**

**[…]**

**In our experiments we implemented the optimisation criteria using ASP *weak constraints* [27]. Describing the underlying technique and precise formulation of the problem is outside the scope of this article; so the reader is referred to our code, available in [23], and [24] for details on the usage of weak rules in CLINGO. For example, conditions of Equations (8a) and (8b) are implemented using two predicates selected/1, holds/1; representing, respectively, the selected constraints among the candidates and their deduction, with the optimisation statements:**

**#minimize{1@2,C: holds(C)}.**

**#minimize{1@1,C: selected(C)}.**

**The selection from the set of candidates is performed by the common *generate and test* ASP paradigm:**

**{ selected(C) : choice(\_,C) }.**

**where choice/2 is the input to the ASP program, encoding the above *sheriffs* function. The *test* part of the paradigm is encoded by constraints enforcing the fact that, for each negative trace, at least one of the constraints in *sheriffs* belongs to the deduction; i.e.**

**rejected(T) :− choice(T,C), holds(C).**

**:− choice(T,\_), not rejected(T).**

Furthermore, to better clarify the computational cost of our technique we added a specific section: 3.4 Complexity considerations.

*And how is the reduction made when many constraints in the original P model are conflicting? It is said that only positive traces are withheld, but what if there are none?*

*Also, the concept of positive and negative is somewhat confusing. It relies on the label of a trace, but there are also accepting/non-accepting traces regarding the constraints. But on page 3, it is mentioned that ‘we only consider as positive the traces allowed by P’. So, the model becomes positive in the sense that there are accepting constraints concerning L+.*

*The experimental evaluation relies on positive traces based on the violation of constraints – so is this a behavioral discrimination, or a label discrimination, or both at the same time (which can happen, e.g., see outcome-oriented predictive process monitoring)?*

We thank the reviewer for the comment as it made us understand that the purposes of our approaches were not sufficiently clear. In short, the overall idea is that we start from two sets of traces L+ and L-, where the criterion used for discriminating between L+ and L- can be whatever kind of criterion (e.g., procedural, related to the trace execution performance or whatever other kind of criteria). Given the two sets, the proposed approach aims at discovering a model that accepts all the traces in L+ and discards as many of the traces in L- as possible. So, as the reviewer pointed out, the criterion for slitting between L+ and L- should be related to the behaviour, because it must be consistent with the fact the traces in L- will be discarded by the emitted model.

In order to clarify this from the beginning, we changed two paragraphs in the introduction as follows.

**Independently of the chosen criterion for splitting the log, we adopt the terms negative and positive example sets to identify the resulting partitioning, keeping in mind that the “negative” adjective is not only connected to unwanted traces, but also to a sort of “upside-down world” of “stranger” behaviours. The information carried by this world diverges from that of the positive examples but—coupled with it—can be used to understand the reasons why differences occur, ultimately providing a more accurate insight of the business process. For this reason, we hereby focus on learning a model that is able to reconstruct which traces belong to which set (by accepting all the positive and rejecting, if possible, all the negative), while reflecting the user expectations on the quality of the extracted model according to predefined metrics.**

Furthermore, our idea also encompasses the possibility to use as a starting point an initial model P that already exists such that it accepts all traces in L+.

Two explanations are needed in this regard:

1) as the reviewer correctly noted, we failed to clarify that if an initial model P is given, then we assume non-conflicting constraints in it.

2) P could have been discovered from the input log with another state-of-the-art technique. In the best scenario, it admits all trace in L+. In practice, sometimes it may happen that P discards some positive traces. As we do not have any control on P (we take it as an input as it is), we assume that the traces in the input log that are not accepted by P are indeed not part of the L+ set. In this sense, we wrote that “we only consider as positive the traces allowed by P”. Nonetheless, we understand that the sentence is confusing if no other explanation is given.

So, we changed the paragraph as follows.

**To apply our technique we only require that the constraints in P are non-conflicting and all the positive traces are compliant with all the constraints in P. \footnote{We are aware that often state-of-the-art approaches do not emit a model compliant with all the traces in the input log. In these cases, we assume that the positive traces in L+ are a subset of the original input including only the traces compliant with P.}**

*Some detailed remarks (please rectify typos):*

*Abstract:*

*- ‘stranger’ behaviours – behaviour singular?*

*- ...results as regards both -> regarding both*

*Section 1:*

*- … as one-class supervised.. -> as a one-class supervised…*

FIXED

*- The link with Stranger Things and the ‘upside-down’ seems a bit much, especially given that the concept of positive vs. negative can be explained in more detail.*

As mentioned above, we changed the paragraph in order to better explain the link and, ultimately, our intentions. We report it again here for clarity.

**Independently of the chosen criterion for splitting the log, we adopt the terms negative and positive example sets to identify the resulting partitioning, keeping in mind that the “negative” adjective is not only connected to unwanted traces, but also to a sort of “upside-down world” of “stranger” behaviours. The information carried by this world diverges from that of the positive examples but—coupled with it—can be used to understand the reasons why differences occur, ultimately providing a more accurate insight of the business process.**

*- …minimal set of constraints that allow… -> that allows*

*Section 2:*

*- …when a trace complete… when a trace completes?*

*- …the a occurs… when activity a or, when a occurs*

*Section 3:*

*- …referred as… referred to as*

FIXED

*- Models and constraints are mixed here. E.g., EXISTENCE(a) is both a model and a constraint in the context of Example 3.1.*

We slightly change the text of the example as follows to correct the error.

**Example 3.1. *The model* M′ = {INIT(a)} *accepts only traces that start with* a*. Hence,* a *exists in each one of those accepted traces. In other words, all those traces also satisfy the model* M = {EXISTENCE(a)}*. However, the latter model also accepts traces that contain* a *even if they do not start with* a *(i.e.,* M ≻ M′*). This relation is valid irrespectively of the involved activity. In a sense, we could say that the template* EXISTENCE(X) *is* more general *than* INIT(X)*.***

*- ..two models going to be compared… with the two models which are going to be compared*

*- …for the easy of understand… for the ease of understanding*

*- Example 3.3 – Let be -> Consider*

*- Theorem 3.1 … the models that are solution… are solutions*

*- Proof - …it would exist an… there exists an… (this appears twice in the proof)*

*- Theorem 3.2 -> Theorem 3.1*

*- …as regards the… as with regard to…*

*- ..ad well.. as well*

*- The Algorithm 1 -> Algorithm 1*

*- Page 6, line 15/16 by Algorithm -> by Algorithm 1*

FIXED

*- …real cases might not fulfil this… Similar to earlier remarks it would help to guide the reader here. Essentially, give a short example mentioning that generated models might not be a perfect positive/negative splitter.*

We reported an example of the cited case:

**Example 3.5. *Consider for example the case of a chosen language bias* D *including* EXISTENCE1(X)*,* EXISTENCE2(X)*,* ABSENCE(X)*,* ABSENCE2(X) *and all binary Declare templates. Let* L+ = {bbb}*,* L− = {bbbb}*. In this case, there is no way to distinguish the input sets because existence1/2 and absence1/2 are not enough, and all binary constraints must be used with two distinct activities.***

*- ..C is then build… -> built*

*Section 4:*

*- …of ‘how good’ is… of ‘how good’ the model is*

*- ..the appraise of the property… appraisal*

*- ..positives from negatives traces… positive/negative*

FIXED

*[1] Fowkes, J., & Sutton, C. (2016, August). A subsequence interleaving model for sequential pattern mining. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 835-844).*

*[2] Egho, E., Gay, D., Boullé, M., Voisine, N., & Clérot, F. (2015, November). A parameter-free approach for mining robust sequential classification rules. In 2015 IEEE International Conference on Data Mining (pp. 745-750). IEEE.*

*[3] De Smedt, J., Deeva, G., & De Weerdt, J. (2019). Mining behavioral sequence constraints for classification. IEEE Transactions on Knowledge and Data Engineering, 32(6), 1130-1142.*

*[4] Slaats, T., Debois, S., & Back, C. O. (2021, September). Weighing the Pros and Cons: Process Discovery with Negative Examples. In International Conference on Business Process Management (pp. 47-64). Springer, Cham.*

*Additional Questions:*

*1. Please explain how this manuscript advances this field of research and/or contributes something new to the literature.: The paper introduces a new process discovery algorithm which explicitly takes into account the label of a trace, i.e., whether it is positive or negative, and mines a minimal model that can discriminate between both classes based on a selected criterion, i.e., generalizability, simplicity, or precision/specificity. It is evaluated on a synthetic case and several real-life case which shows its effectiveness.*

*2. Is the manuscript technically sound? Please explain your answer under Public Comments below.: Yes*

*1. Which category describes this manuscript?: Research/Technology*

*2. How relevant is this manuscript to the readers of this periodical? Please explain your rating under Public Comments below.: Very Relevant*

*1. Are the title, abstract, and keywords appropriate? Please explain under Public Comments below.: Yes*

*2. Does the manuscript contain sufficient and appropriate references? Please explain under Public Comments below.: Important references are missing; more references are needed*

*3. Does the introduction state the objectives of the manuscript in terms that encourage the reader to read on? Please explain your answer under Public Comments below.: Yes*

*4. How would you rate the organization of the manuscript? Is it focused? Is the length appropriate for the topic? Please explain under Public Comments below.: Satisfactory*

*5. Please rate the readability of the manuscript. Explain your rating under Public Comments below.: Easy to read*

*6. Should the supplemental material be included? (Click on the Supplementary Files icon to view files): No, it should not be included at all*

*7. If yes to 6, should it be accepted: After revisions. Please include explanation under Public Comments below.*

*Please rate the manuscript. Please explain your answer.: Good*

Referee #3

*Recommendation: Author Should Prepare A Major Revision For A Second Review*

*Comments:*

*In this paper, the authors focus on declarative processes and embrace the less-popular view of process discovery as a binary supervised learning task, where the input log reports both examples of the normal system execution, and traces representing “stranger” behaviors. They have also tested their approaches on several toy and benchmark data sets. I think the problem itself is interesting and the following comments should be considered.*

*(1) The first is about the novelty. I think that the paper provides a new view and investigate this problem by declarative processes. But the novelty cannot be accessed clearly. I think just a new view without plenty of support is not enough.*

The authors wish to thank the reviewer for all the precious comments. He is right that we probably failed in clarifying the novelty of our approach. To address this issue, we expanded the description of our contributions and put it into a dedicated section (1.1).

**In the framework described so far, our work advances the state of the art by proposing the following contributions.**

**• A novel discovery approach, NegDis, based on the underlying logic semantics of Declare, which makes use of the information brought by the positive and negative example sets to produce declarative models. The resulting technique can be used to either mine Declare process models from scratch, or refine an existing model with additional knowledge.**

**• The adoption of a satisfiability-based technique to identify the models.**

**• A heuristic to select the models according to the user preferences of generalisation or simplicity, where the language bias and the subsumption rules between Declare templates are not hard-coded in the discovery algorithm, but provided as an input for a finer-grain configurability.**

**• An evaluation of the performance of NegDis w.r.t. other relevant works in the same field, highlighting strengths and weaknesses of our technique.**

*(2) The second is about the methodology. I think all the deductions are based on logic programming. Do these deductions have theoretical support or they are all empirical? Please make a declaration.*

We agree that the implementation details reported in the paper were not enough to clarify this point, so we added some paragraphs in section 3.3 that we report here:

**The second optimisation stage has been implemented using the Answer Set Programming (ASP) system CLINGO [24]. The main reason for selecting an ASP system for finite domain optimisation is that rules provide an effective and intuitive framework to implement a large class of closure operators. Indeed, all the deductive systems for Declare that we analysed in the literature (see e.g. [22], [25]) are expressed in the form of logic rules c1∧ . . . ∧ cn =⇒ cn+1, where ci are Declare constraints or their negation; therefore, they can be equivalently expressed as Normal Logic Programs (NLPs) [26] by exploiting the assumption that the set of activities is finite and known in advance. The declarative semantics of NLPs ensures that the (unique) deductive closure of the rules is taken into account for the optimisation stage. For example the valid formula INIT(a) ∧ b!=a ⇒ PRECEDENCE(a,b) that holds for any pair of activities a, b can be written as the rule**

**precedence(A,B) :− init (A) , activity (B) , A != B.**

**using a specific predicate (activity/1) representing the set of all activities. \footnote{** **By leveraging CLINGO function symbols, Declare constraints are encoded as function terms; for example, the constraints derived are represented using the unary predicate holds/1: holds(precedence(a1,a2)) }**

**[…]**

**In our experiments we implemented the optimisation criteria using ASP *weak constraints* [27]. Describing the underlying technique and precise formulation of the problem is outside the scope of this article; so the reader is referred to our code, available in [23], and [24] for details on the usage of weak rules in CLINGO. For example, conditions of Equations (8a) and (8b) are implemented using two predicates selected/1, holds/1; representing, respectively, the selected constraints among the candidates and their deduction, with the optimisation statements:**

**#minimize{1@2,C: holds(C)}.**

**#minimize{1@1,C: selected(C)}.**

**The selection from the set of candidates is performed by the common *generate and test* ASP paradigm:**

**{ selected(C) : choice(\_,C) }.**

**where choice/2 is the input to the ASP program, encoding the above *sheriffs* function. The *test* part of the paradigm is encoded by constraints enforcing the fact that, for each negative trace, at least one of the constraints in *sheriffs* belongs to the deduction; i.e.**

**rejected(T) :− choice(T,C), holds(C).**

**:− choice(T,\_), not rejected(T).**

Also, we want to remark that in this work, we did not define the subsumption rules employed in our deductive closure operator, but we took them from other works dedicated to the point, like [20] where the authors provide a subsumption hierarchy of templates and a demonstration of the regular expressions to represent each template.

*(3) The third is about experimental results. The authors have only employed three data sets with their variants. I think it is far from enough. Plenty of experiments are required for support, especially for the data mining community.*

Thank you for the suggestion. We have included the results of other tests and their discussion in section 4.

One of the main problems of the BPM and process mining community is indeed the availability of real-life logs with negative examples. So, to increase the number of tests, beside including three additional datasets, we experimented with different, artificially created labelling. In particular, for each dataset we provide a domain-specific labelling that is related to the behaviour of the traces (e.g., for DREYERS dataset we classify the executions based on whether they were reset due to a system failure or not), and two other labelling based on the mean and median duration of the traces.

As a consequence, the number of the different datasets in the evaluation is now 18. Each one is used to assess the performance of both our approach and that of DeclareMiner[32].

*(4) The presentations should be improved and a native speaker is suggested for the presentation.*

We did our best to improve the presentation. A careful proofread highlighted several typos and errors that we promptly corrected in the current version.

*Additional Questions:*

*1. Please explain how this manuscript advances this field of research and/or contributes something new to the literature.: See the details below.*

*2. Is the manuscript technically sound? Please explain your answer under Public Comments below.: Appears to be - but didn't check completely*

*1. Which category describes this manuscript?: Research/Technology*

*2. How relevant is this manuscript to the readers of this periodical? Please explain your rating under Public Comments below.: Relevant*

*1. Are the title, abstract, and keywords appropriate? Please explain under Public Comments below.: Yes*

*2. Does the manuscript contain sufficient and appropriate references? Please explain under Public Comments below.: References are sufficient and appropriate*

*3. Does the introduction state the objectives of the manuscript in terms that encourage the reader to read on? Please explain your answer under Public Comments below.: Could be improved*

*4. How would you rate the organization of the manuscript? Is it focused? Is the length appropriate for the topic? Please explain under Public Comments below.: Could be improved*

*5. Please rate the readability of the manuscript. Explain your rating under Public Comments below.: Readable - but requires some effort to understand*

*6. Should the supplemental material be included? (Click on the Supplementary Files icon to view files): Does not apply, no supplementary files included*

*7. If yes to 6, should it be accepted:*

*Please rate the manuscript. Please explain your answer.: Good*