

BUSINESS PROCESS MINING USING ITERATED LOCAL SEARCH

DIANA CHINCES⁽¹⁾ AND IOAN SALOMIE⁽¹⁾

ABSTRACT. This paper introduces ILS Miner, a novel method for discovering process models from event logs using iterated local search. A comparison between ILS Miner, GLS Miner and HeuristicMiner is presented in this paper. GLS Miner was chosen because applies Guided Local Search for discovering process models from event logs, HeuristicMiner[10] being a legacy method in business process discovery. It is shown that ILS Miner can discover process models that correctly map to the event log. ILS Miner works with business processes represented as graphs, and the final discovered process is represented as a BPMN diagram.

1. INTRODUCTION

The work efficiency in large organizations is given by optimal process flows that are followed in the company. In some cases, these process flows are just defined and followed by employees, but in most of the cases the process flows (models) need to be discovered. Nowadays, organizations use different management systems that keep track of all the work being done, generating so called event logs. The process model that is being followed in the organization can be discovered from these event logs using process mining algorithms. Figure 1 shows an overview of process mining. The event logs are generated by the information system used in the company and are the input for the process discovery algorithms. Not all the information from the database is included in the event logs. In [3] the following assumptions are defined: each event refers to an activity (well-defined step in the process), each event refers to a case (the

Received by the editors: April 13, 2013.

2010 *Mathematics Subject Classification.* 68-00, 68M14, 68M01, 68N01, 68P10 .

1998 *CR Categories and Descriptors.* K.4.3 [**Organizational Impacts**]: *Reengineering*; K.4.3 [**Organizational Impacts**]: *Automation*; K.4.3 [**Organizational Impacts**]: *Computer-supported collaborative work*.

Key words and phrases. Guided Local Search, GLS Miner, Iterated Local Search, business process mining, event log, BPMN .

This paper has been presented at the International Conference KEPT2013: Knowledge Engineering Principles and Techniques, organized by Babeș-Bolyai University, Cluj-Napoca, July 5-7 2013.

process instance), each event has a performer, and events have a timestamp and are totally ordered. Knowing this, an event can be defined as a tuple (c, a, p) , where c is the case, a the activity and p the performer. The events being ordered in time, causal relations can be defined between the activities and the actors. Given two tuples (c, a_1, p_1) and (c, a_2, p_2) from the event logs, it results that activity a_1 that is performed by person p_1 is followed by activity a_2 , which is performed by person p_2 . For applying Guided Local Search for process discovery the performer is not needed, the algorithm only makes use of the cases, the activities and their order.

Three different perspectives of process mining were identified in [4]: the pro-

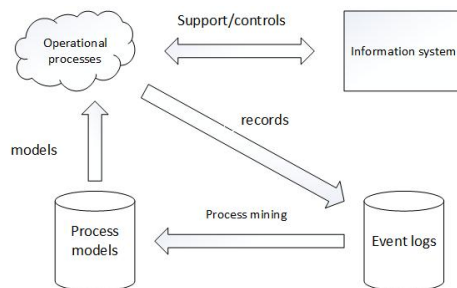


FIGURE 1. Overview of process mining[2]

cess perspective, the organizational perspective and the case perspective. The process perspective deals with discovering the process models from the event logs. Areas of applicability of the process perspective include IT management processes governing the operations of services and infrastructure as well as scientific computing workflows. This process model can be further used as input for applying optimization algorithms for improving the workflows in the organization. The organization perspective focuses on the relations between the individuals in the company and the case perspective mines the properties of the cases in the process. An important property of a case is the data attached to it. For example, if a case represents submitting an order it would be interesting to have knowledge of the products ordered, the quantity, prices paid, discounts etc.

This paper aims at presenting a novel method for discovering process models from event logs using Iterated Local Search, focusing on the process perspective.

The remainder of the paper is organized as follows. Section 2 presents the background of process mining and related work. Section 3 explains the Iterated Local Search algorithm. Section 4 describes how Iterated Local Search is used for process discovery, followed by Section 5 where the experiments

of applying ILS Miner in process discovery are presented. In Section 6 the conclusions of this paper are stated.

2. BACKGROUND AND RELATED WORK

This section reviews the state of the art in business process discovery and describes a common structure used for storing event logs for process mining. The information systems presented in Figure 1 use different data structures to store the data. In [5] a generic framework for process discovery algorithms and a common structure to be used were introduced. The ProM framework allows the creation of plugins for the process mining algorithms. The input for a plugin is an XML file in the MXML format [5]. Tools like Open Xes [6] allow the mapping and conversion between different log storing structures and MXML files.

Process mining algorithms, such as the Alpha algorithm [8] assume that it is possible to sequentially order events, such that each event refers to a case and an activity. The Alpha algorithm does not take into consideration the timestamp of events. This information is used to automatically construct a process model, which in the Alpha algorithm is represented as a Petri Net describing the event logs.

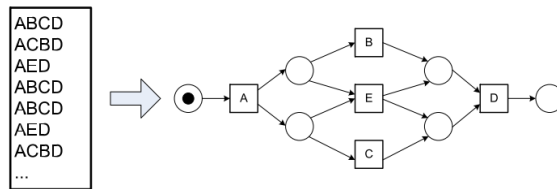


FIGURE 2. The event log and the process model discovered by Alpha algorithm [9]

In Figure 2 a simple event log and the Petri-Net that was generated by applying the Alpha algorithm can be observed. It can be noted from the figure that the event log is easily reproduced from the discovered model.

The GLS Miner[14] is the first algorithm in which local search is applied in the business process domain. It is shown in [14] that the GLS Miner can correctly discover heterogenous types of business processes. The search space for applying guided local search is a set of processes, represented as graphs. A process model is represented as a graph. The nodes of the graph represent the existent activities in the process model and the edges the flows between the activities. The features necessary for guided local search were defined as

all possible edges in the graph. Thus, a solution has a feature available if and only if the specific edge is available in the solution. The algorithm starts with a random graph and can either add or remove edges in order to move from a solution to another.

The HeuristicMiner [10] is a heuristic driven process mining algorithm, being considered one of the most relevant process discovery algorithms. It is a practical applicable mining algorithm that can deal with noise (e.g. exceptions from the usual workflow that appear in the event log) and can be used to express the main behavior registered in an event log. The HeuristicMiner only considers the order of the events within a case by using the timestamp of the activities. A log is defined in the same way as for the Alpha algorithm [9]. The following log will be used for explaining the HeuristicMiner, $W = [ABCD, ABCD, ACBD, ACBD, AED]$.

HeuristicMiner builds a dependency graph [10]. A frequency based metric is used to indicate how certain we are that there is truly a dependency relation between two events A and B:

$$a \Rightarrow_W b = \left(\frac{|a >_W b| - |b >_W a|}{|a >_W b| + |b >_W a| + 1} \right)$$

A high value of the \Rightarrow_W relation between a and b determines that there is dependency relationship between a and b. Applying the heuristics to the event log in Table 1 takes us to the results in Figure 3.

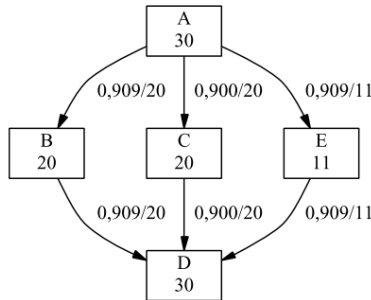


FIGURE 3. Dependency graph [10]

3. ITERATED LOCAL SEARCH

Iterated local search can be easily explained as [12]: one iteratively builds a sequence of solutions generated by the embedded heuristic, leading to better

solutions than if one were to use repeated random trials of that heuristic. There are two main points that make an algorithm an iterated local search: (i) there must be a single chain that is being followed (thus excluding population-based algorithms); (ii) the search for better solutions occurs in a reduced space defined by the output of a blackbox heuristic.

A local search is a problem specific heuristic optimization algorithm. In most cases this algorithm can be improved iteratively, resulting in an iterated local search algorithm. Algorithm 1 shows the generic iterated local search algorithm.

```

begin
   $s_0 \leftarrow \text{GenerateInitialSolution};$ 
   $s^* \leftarrow \text{LocalSearch}(s_0);$ 
  while not termination criterion do
     $s' \leftarrow \text{perturbation}(s^*, \text{history});$ 
     $s'^* \leftarrow \text{LocalSearch}(s');$ 
     $s^* \leftarrow \text{AcceptanceCriterion}(s^*, s'^*, \text{history});$ 
  end
end

```

Algorithm 1: Iterated Local Search Algorithm [12]

Iterated local search runs for each problem until the termination criteria is met. The termination criteria has to be created for each particular problem. The iterated local search walk is not reversible. First, an initial solution is generated on which the search algorithm is applied. Until the termination criterion is met, the current best solution is perturbed and if it meets the acceptance criterion it may be used as the next best solution. A history of perturbations can be included in the algorithm, so that the same perturbation is not applied several times on the same solution.

4. ITERATED LOCAL SEARCH MINER

From the previous section presenting the general Iterated Local Search algorithm, we can conclude that for applying this for a specific problem, the following items need to be defined:

- a termination criterion
- a method for perturbation
- an acceptance criterion

The search space for applying iterated local search is a set of processes, represented as graphs. The nodes of the graph represent the existing activities in the process model while the edges the flows between the activities. Algorithm 2 shows the Iterated Local Search Miner (ILS Miner).

```

input : event log, maxIterations
output: process model mapped on the event log
begin
  solution  $\leftarrow$  random(log);
  currentCost  $\leftarrow$  computeCost(solution);
  neighbor  $\leftarrow$  findNeighbor(solution);
  k  $\leftarrow$  0;
  while neighbor  $\neq$  null and k < maxIterations do
    solution  $\leftarrow$  neighbor;
    perturb(solution);
    second  $\leftarrow$  findNeighbor(solution);
    currentCost  $\leftarrow$  computeCost(solution);
    secondCost  $\leftarrow$  computeCost(second);
    if secondCost j currentCost then
      | solution  $\leftarrow$  second
    end
    k  $\leftarrow$  k + 1;
  end
  returnsolution;
end

```

Algorithm 2: ILS Miner

We considered the termination criterion being the fact that the algorithm cannot find a better neighbor and the number of maximum iterations have been executed. Algorithm 3 shows the perturbation procedure.

```

input : graph
output: graph'
begin
  edge  $\leftarrow$  randomEdge();
  graph.add(edge);
  return graph
end

```

Algorithm 3: Perturbation algorithm

In the perturbation algorithm the graph is randomly modified. The algorithm perturbs the graph by adding a random edge to it. The perturbed solution and its neighbor is considered to pass the acceptance criterion if it has a lower cost than the previous best solution. We have chosen to compute the cost as the sum of the frequency of edges in the graph. The frequency of an edge (k,s) is computed based on the event log given as input to the mining algorithm. The frequency of the occurrence (k,s) is computed as

$$(1) \quad \frac{(\text{number of occurrences of } (k,s))}{(\text{number of occurrences of } (s,k) + 1)}$$

5. EXPERIMENTS AND RESULTS

The Iterated Local Search BP Miner was implemented as a ProM Framework[5] plugin. This plugin was created for proof of concept of the algorithm described in the previous section.

The results are presented as a comparison between the proposed Iterated Local Search BP Miner, the Guided Local Search BP Miner[14] and the Heuristic Miner[10]. This comparison is done considering the discovered process models and the errors encountered by each of the algorithms. Process models P1 (Figure 4), P2 (Figure 5) and P4 (Figure 6) were considered for this comparison. P1 is a simple process model, having just a few flows from the start to the end activity. P2 is more complex, having alternative flows from the start to the end activity. P3 has the most complexity as it contains loops. We considered for the experiments these three basic types of processes for prove of concept. In the future, we plan to run scalability tests on the algorithm using event logs from real data. Table 1 shows the event logs that were used for each process. The regularization value used for the experiments was 0.6, however this is easily customizable in the ProM Framework plugin.

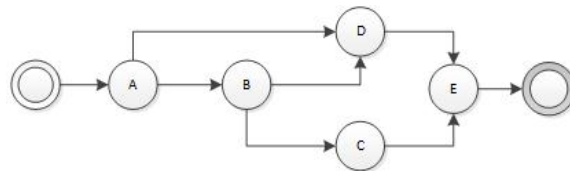


FIGURE 4. P1 - simple process model

Table 2 shows the comparison of the discovered process models by applying the ILS BP Miner and the GLS BP Miner. For process model P1 the ILS BP Miner, as well as the GLS Miner, have correctly discovered the model from the given event logs. We can observe that the Guided Local Search Miner has discovered a process model that correctly maps to the event log, even if this

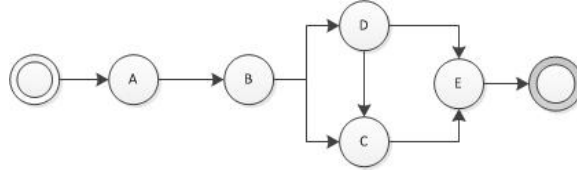


FIGURE 5. P2 - simple process model with alternative paths

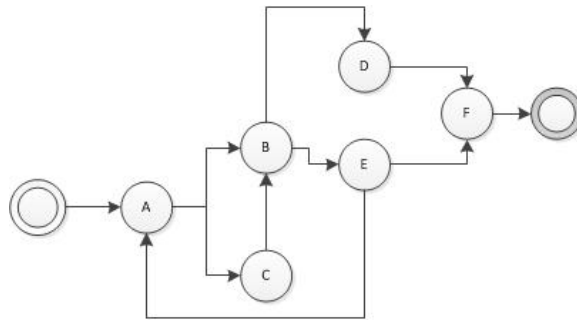


FIGURE 6. P2 - process model with loops

is not the same with the initial process model. Applying a process mining algorithm in an organization might end up with this kind of results, that the process model that is discovered from the event logs is not exactly the same with the process model that was initially defined in the organization. For process model P2 the both algorithms have correctly discovered the initial model, as shown from the error graph as well. For the last process model, the GLS Miner did not discover the entire model, but managed to obtain a nearly correct process model. ILS Miner however discovered the a model which contains less errors than the model discovered by the GLS Miner.

Table 3 shows the comparison of the discovered process models by applying the ILS BP Miner and the Heuristic Miner[10]. For process model P1 the ILS BP Miner, as well as the GLS BP Miner, have correctly discovered the model from te given event log. We can observe that for process model P2, ILS BP Miner has correctly discovered the process model, but the Heuristic Miner is missing the edge from **D** to **E**. For the last process model, ILS BP Miner has managed to obtain once again a model that is better than the Heuristic Miner. This can be observed also from Figure 6, showing the number of errors encountered by each of the algorithms.

Figure 7 shows the number of errors encountered by each of the three algorithms that were experimented.

Traces for P1			Traces for P3		
Case	Activity	Time			
			1	A	19:00:00
1	A	19:00:00		B	19:01:00
	B	19:01:00		D	19:02:00
	D	19:02:00		F	19:03:00
	E	19:03:00	2	A	19:00:00
2	A	19:00:00		B	19:01:00
	C	19:01:00		E	19:02:00
	E	19:02:00		F	19:03:00
3	A	19:00:00	3	A	19:00:00
	B	19:01:00		C	19:01:00
	D	19:02:00		B	19:02:00
	E	19:03:00		E	19:03:00
Traces for P2				F	19:04:00
1	A	19:00:00	4	A	19:00:00
	B	19:02:00		C	19:01:00
	C	19:03:00		E	19:02:00
	E	19:04:00		F	19:03:00
2	A	19:00:00	5	A	19:01:00
	B	19:01:00		B	19:02:00
	D	19:02:00		E	19:03:00
	C	19:03:00		A	19:04:00
	E	19:04:00		B	19:05:00
3	A	19:00:00		D	19:06:00
	B	19:01:00		F	19:07:00
	D	19:02:00	6	A	19:01:01
	E	19:03:00		C	19:02:00
				B	19:03:00
				E	19:04:00
				F	19:05:00

TABLE 1. Traces considered for experiments

6. CONCLUSION

This paper introduced ILS Miner, a new algorithm for discovering process models from event logs. The experiments were done on heterogenous types of processes and the results were compared with GLS Miner[14], another algorithm that applies local search for process discovery and HeuristicMiner[10], a legacy algorithm in process discovery. This paper shows that ILS Miner has

Process model	Initial process (known)	ILS BP Miner	GLS BP Miner
P1			
P2			
P3			

TABLE 2. The initial (known) process model and the discovered models using ILS Miner and GLS Miner

Process model	Initial process (known)	ILS BP Miner	Heuristic Miner
P1			
P2			
P3			

TABLE 3. The initial (known) process model and the discovered models using ILS Miner and Heuristic Miner

managed to correctly discover the process models from the event logs, showing that the results are better than the ones obtained by both of GLS Miner and Heuristic Miner. In the future, we plan to apply our methods on more complex process models and on real data, comparing the results by the performance of the algorithms.

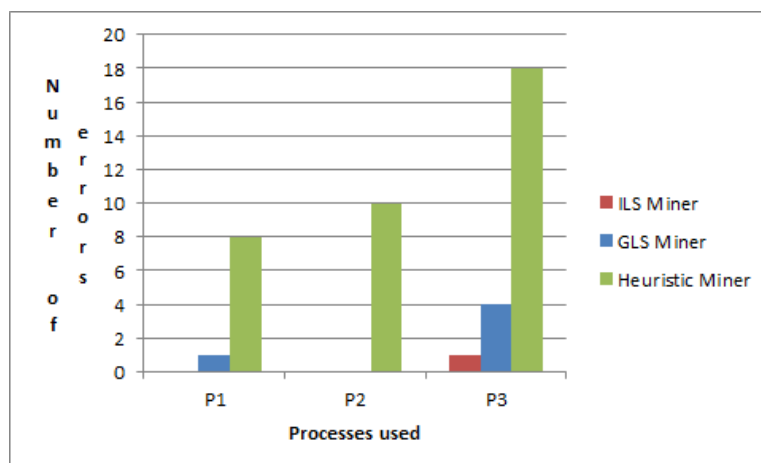


FIGURE 7. Number of errors encountered by the algorithms

REFERENCES

- [1] Chris Voudouris, Edward Tsang, *Guided Local search*, Technical Report CSM-247, Department of Computer Science, University of Essex, August 1995
- [2] A.K. Alves de Medeiros, A.J.M.M. Weijters and W.M.P. van der Aalst, *Genetic Process Mining An Experimental Approach*, Data Mining and Knowledge Discovery Volume 14 Number 2 (2007), pages 245-304
- [3] W.M.P. van der Aalst, A.K.A. de Medeiros, *Process Mining and Security: Detecting Anomalous Process Executions and Checking Process Conformance*, Second International Workshop on Security Issues with Petri Nets and other Computational Models (WISP 2004), pages 6984
- [4] W.M.P. van der Aalst, *Business Alignment: Using Process Mining as a Tool for Delta Analysis and Conformance Testing*, Requirements Engineering Volume 10 Number 3 (2005), pages 198-211
- [5] B.F. van Dongen and W.M.P. van der Aalst, *A Generic Import Framework for Process Event Logs*, J. Eder and S. DÜstard, editors, Business Process Management Workshops, Workshop on Business Process Intelligence (BPI 2006), volume 4103 of Lecture Notes in Computer Science, pages 81-92
- [6] Open Xes Homepage, <http://xes-standard.org/openxes/start>
- [7] Christos Voudouris, Edward Tsang, *Guided Local Search and its application to the traveling salesman problem*, European Journal of Operational Research 113 (1999), pages 469-499
- [8] W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster. *Workflow Mining: Discovering Process Models from Event Logs*. IEEE Transactions on Knowledge and Data Engineering, 16(9):1128-1142, 2004
- [9] W.M.P. van der Aalst, V. Rubin, B.F. van Dongen, E. Kindler, and C.W. Gunther, *Process Mining: A Two-Step Approach using Transition Systems and Regions*, Software and System Modeling Volume 9 Number 1 (2010) pages 87-111

- [10] A.J.M.M. Weijters, W.M.P. van der Aalst, and A.K. Alves de Medeiros, *Process Mining with the Heuristics Miner Algorithm*, BETA Working Paper Series, WP 166, Eindhoven University of Technology, Eindhoven
- [11] W.M.P van der Aalst, *Business Alignment: Using Process Mining as a Tool for Delta Analysis and Conformance Testing*, Requirements Engineering Volume 10 Number 3 2005, pages 198-211
- [12] Iterated Local Search, <http://ideas.repec.org/p/upf/upfgen/513.html>
- [13] Helena R. Lourenco, Olivier C. Martin, Thomas Stutzle, Iterated Local Search, *Handbook of Metaheuristics*, F. Glover and G. Kochenberger, Eds. *International Series in Operations Research & Management Science*, vol. 57. Kluwer Academic Publishers, Norwell, MA, 321353
- [14] Diana Chinces, Ioan Salomie, Business Process Mining using Guided Local Search, *The 12th International Symposium on Parallel and Distributed Computing - ISPDC 2013* (accepted)

⁽¹⁾ TECHNICAL UNIVERSITY OF CLUJ-NAPOCA
E-mail address: `diana.chinces@cs.utcluj.ro`

E-mail address: `ioan.salomie@cs.utcluj.ro`