Advances in Computational Sciences and Technology
ISSN 0973-6107 Volume 10, Number 6 (2017) pp. 1607-1614

http://www.ripublication.com

Non-Dominated Bi-Objective Genetic Mining Algorithm

Sonia[1], Aarti Purwaha[1] and Aashu Rani[1]

[1]Department of Computer Science, University of Delhi, New Delhi, India

Abstract

Process mining generates process model from event logs and used to connect data mining techniques to process modelling, analysis, simulation etc. Many process mining techniques are used to generate model all create a single model. In this work bi-objective genetic process mining algorithm has been used to generate multiple models also called as non-dominated solutions. All of these models are equally good.

Keywords: Process models, Petri-nets, Process discovery, Optimization, Genetic Mining.

1. INTRODUCTION

Data generated from real world processes are stored in the form of event logs [3]. Event log is a collection of traces where each trace is a sequence of events and each event has an activity, a case, and the time at which the event occurred. Process models are in the form of BPMN, Petri-nets, EPC, or UML diagram. Most widely used graphical notation is Petri-net containing a place, an activity, and arc which shows the precedence, concurrency, or occurrence of events [8]. There are three types of process mining namely: Process discovery [2], the most prominent technique which discovers model from an event log with no a-priori information; Conformance checking, which shows the deviations between process model and the event log and Enhancement technique that is used to improve the process model with the help of information present about the process in the event log [1].

There are various perspectives being focused upon by process mining. The control-flow perspective discovers a combination of all possible paths having order of activities. The organizational perspective is the discovery of social networks, resource

behavior and organizational structures. It also focuses on the hidden information (i.e., which actors are involved and how are they related) present in the log. The case perspective is concerned with properties of cases (i.e., values of corresponding data elements, path in the process or actors working on the process). The time perspective focuses on the timing and frequency of events. With existence of event log where each event refers to a case, activity and a point in time, it is possible to discover bottlenecks, monitor the utilization of resources, predict the remaining processing time of running cases, and measure service levels.

Instead of being a specific type of data mining, process mining is also a missing link between traditional model-driven BPM and data mining. Traditional data mining techniques are not process centric and do not address process discovery, conformance checking, and enhancement. Concurrency and end-to-end process models are important for process mining. Process mining is useful for both offline and online analysis. For example completion time of an event can be predicted using discovered process models.

2. PROCESS DISCOVERY ALGORITHMS

There are various process discovery algorithms [2]; Deterministic mining algorithms produce the same result for same input by considering the timestamp and calculate the control flow of events in the log (i.e. α-Algorithm, α^+-Algorithm). Heuristic mining [4] incorporates frequencies of traces and events with deterministic mining to reconstruct a process model. In this algorithm model produced from highly complex real processes are very complex and hard to understand, this algorithm neglects the infrequent paths while constructing a process model. Genetic mining algorithm [9] is not deterministic and uses the natural evolutionary approach by generating a random population of process models. The solution is produced in four steps (initialization, selection, reproduction and termination). The individuals are iteratively selected from the population and reproduced using crossover and mutation over different generations. Better fitted models are generated due to selection and reproduction.

Various process mining tools [1] are used to generate models from the event log such as ProM, Disco, RapidProM etc. ProM supports a wide variety of plug-ins and aims to cover whole process mining spectrum. It supports many notations such as petri-nets, transition systems, C-nets, fuzzy models, BPMN, Declare, etc. This tool is very powerful, extendible, and supports conformance checking and operational support. Disco has powerful filtering capabilities for comparative process mining and adhoc checking of patterns. This tool focuses on discovery and performance analysis including animation and uses the variant of fuzzy models. This tool does not support conformance checking and operational support. RapidProM can be used to combine process mining with data mining, text mining etc. as it is an extension of ProM

framework with RapidMiner. This tool can be used to repeat the process mining analyses (scientific experiments, reusable macros).

The quality of process model is based on the four quality dimensions (Generalization, Completeness, Simplicity, and Preciseness) [9]. Completeness tells that the traces in the event log can be reproduced with the mined model. Preciseness is used to describe under fitted behavior in log while generalization avoids overfitting. Simplicity describes the complexity of the model. All these four qualities are equally important to find a good process model. Completeness is necessary to find the fitness of a model so we use it as one of the measure to make sets for optimizing the objectives namely completeness-generalization, completeness-simplicity, and completeness-preciseness.

In this work we have used multi-objective genetic mining optimization algorithm to find the Pareto front solutions [5]. The models generated from multi-objective algorithms are equally good. There are many evolutionary algorithms such as NSGA, PAES, SPEA, SPEA-II but we used NSGA-II to find mutually non-dominated solutions as NSGA-II is more efficient than other algorithms [6] and SPEA-II takes longer time for large event logs.

3. METHODOLOGY

Initial population was randomly generated and evaluated to find the causal relations between different activities in an event log, then Non-dominated sorting algorithm [6] [7] (see algorithm 1) is used to find the individuals to which any individual (i) dominates and the number of individuals that dominate this individual (i).

Algorithm 1 Non-Dominated Sorting

```
1: Total size=Set of individuals from which we need to find non-dominated solutions
   1
2: for i=1 : Total size do
3:     for j=1 : Total size do
4:         if Completeness(i) >= Completeness(j) AND Generalization(a1) >=
   Generalization(a2) then
5:             s ← Individuals to which this individual i dominate
6:         else if Completeness(i) <= Completeness(a2) AND
   Generalization(a1) <= Generalization(a2) then
7:             n ← Number of individuals that dominate the individual i
8:         end if
9:         if n==0 then
10:            front ← the individual i which is not being dominated by any other individual
11:        end if
12:    end for
13: end for
14: Pareto-front solutions
```

Algorithm 1: Non-Dominated Sorting Algorithm

Then we select the individuals on the basis of binary tournament selection (see algorithm 2) and selected individuals under goes cross-over and mutation and then this mutated population is combined with the original population to give the best individuals(Elitism) for the next iteration. After every iteration, first front stores the individuals which are non-dominated (explained in Fig.1).

Algorithm 2. Multi-objective Binary Tournament Selection

```
1: 2N individuals, as each individual play twice
2: if Completeness(a1) >= Completeness(a2) AND Generalization(a1) >=
   Generalization(a2) then
3:     dominating individual a1 wil go to selection pool
4: else if Completeness(a1) <= Completeness(a2) AND Generalization(a1) <=
   Generalization(a2) then
5:     dominating individual a2 wil go to selection pool
6: else
7:     Non-Dominated Pair
8:     if Completeness(a1) >= Completeness(a2) then
9:         a1 will go to selection pool
10:    else
11:        a2 will go to selection pool
12:    end if
13: end if
14: Best N individuals in selection pool
```

Algorithm 2: Binary Tournament Selection

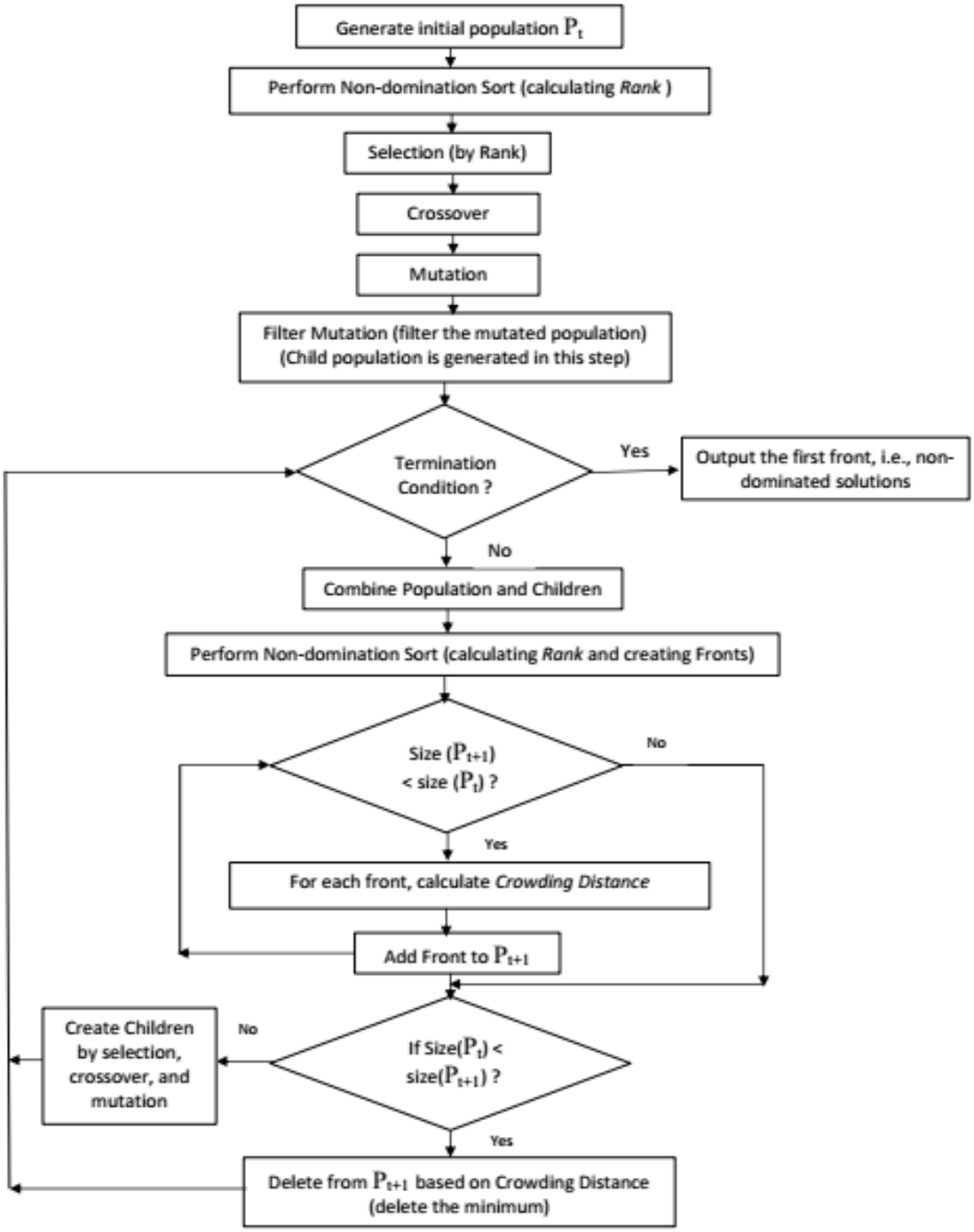

Fig.1: NSGA-II bi-objective genetic mining algorithm.

4. RESULTS

The algorithm was run in matlab-2014a for 100 iterations and for a population size of 100. We used data from real business processes BPI2013

(http://www.win.tue.nl/bpi/doku.php?id=2013:challenge). This data set contains 13 activities, 65533 events and 7554 traces. As the number of activities increases in the log time required for convergence of the algorithm increases. For large event logs it is very hard to see the relation between different activities. Process mining helps in analyzing the data with the help of generated model and NSGA-II generates more than one model so user can choose out of them. Fig. 2, 3 and 4 shows that the non-dominated solutions 16, 5 and 9 are obtained from bi-objective sets (completeness-simplicity, completeness-preciseness, and completeness-generalization resp.)

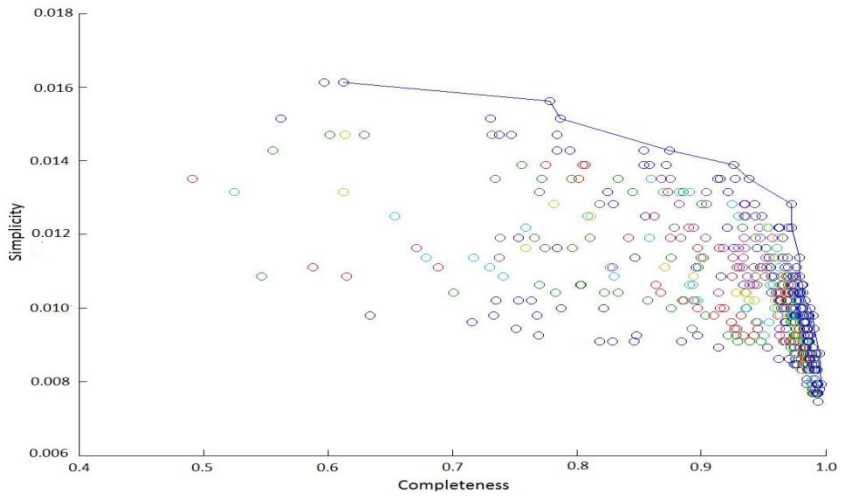

Fig. 2: Non-dominated Solutions with objectives Completeness-simplicity

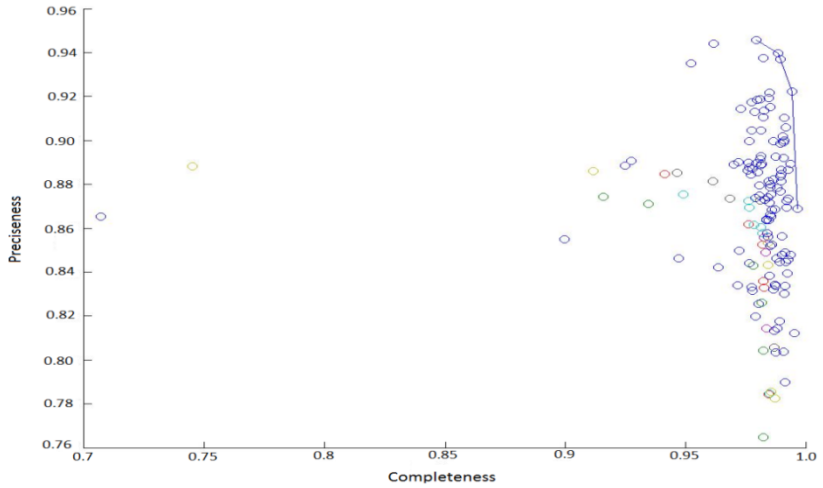

Fig. 3: Non-dominated Solutions with objectives Completeness-Preciseness

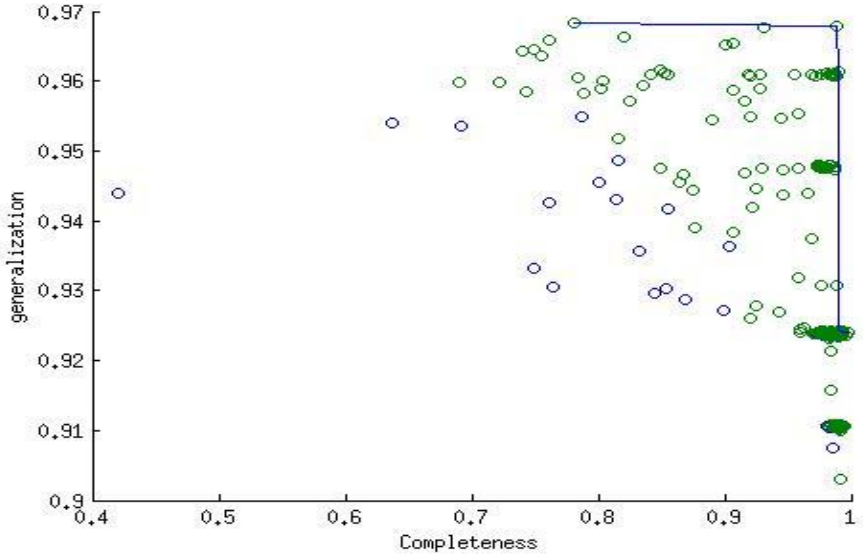

Fig. 4: Non-Dominated Solutions with objectives Completeness-generalization

5. CONCLUSION

The experimentation has demonstrated the effectiveness of the multi-objective evolutionary approach. According to the user requirement based on the features the algorithm produces more than one model, so the user has option to select from the models which are equally good. But we face many limitations and challenges while discovering process models for the event logs involving large volumes and high velocity of event log data as the process discovering task becomes computationally intensive. So in future we want to parallelize process mining algorithms.

REFERENCES

[1] Van der Aalst, Wil MP. Process mining: data science in action. Springer, 2016

[2] J.E. Cook, A.L. Wolf: Discovering models of software processes from event-based data. ACM Trans. Softw. Eng. Methodol. (TOSEM) 7 (3), 215249, 1998.

[3] Wil Van der Aalst, Ton Weijters, and Laura Maruster: Work-flow mining: Discovering process models from event logs. Knowledge and Data Engineering, IEEE Transactions on, 16(9):1128-1142, 2004.

[4] AJMM Weijters, Wil MP van Der Aalst, and AK Alves De Medeiros: Process mining with the heuristics miner-algorithm. Technische Universiteit Eindhoven, Tech. Rep. WP, 166:1-34, 2006.

[5] K. Deb: Multi-Objective Optimization. In E.K. Burke and G. Kendall, editors,

Search Methodologies, pages 273-316. Springer US, 2005.

[6] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. Lecture notes in computer science, 1917:849-858, 2000.

[7] N. Srinivas and K. Deb: Multi-objective optimization using non-dominated sorting in genetic algorithms, Evolutionary Computation, vol. 2, no. 3, 221248, Fall 1994.

[8] T. Murata: Petri nets: properties, analysis and applications, Proc. IEEE 77 (4), 541580, 1989.

[9] A. de Medeiros: Genetic Process Mining, Ph.D. thesis, Technische Universiteit Eindhoven, 2006.