A Cloud Theory-based Simulated Annealing for Discovering Process Model from Event Logs

Mirpouya Mirmozaffari [1], Mostafa Zandieh [2] and Seyed Mojtaba Hejazi [3]

[1,3]Msc. Student, Department of Industrial Engineering, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran (Email:m.mirmozaffari@gmail.com)

[2] Associate Professor, Department of Industrial Management and accounting, Faculty Shahid Beheshti University, G.C. Tehran, Iran (Email: m_zandieh@sbu.ac.ir)

***Abstract*:** *One of the most important areas of process mining studies is the discovery of the process model. Better convergence in the process discovery by using metaheuristic algorithms is an important issue and various studies have been done in this field. In this paper, a Cloud theory-based simulated annealing is offered in which a casual matrix and a new function are used to display and evaluate the answer respectively. By designing two different tests, the performance of this algorithm with simulated annealing has been investigated using statistical hypothesis tests. The results illustrate the superior performance of CSA algorithm to SA algorithm.*

Keywords: *Process mining, Process discovery, Casual matrix, Simulated Annealing, Cloud theory-based simulated annealing.*

1. Introduction

Process mining is a new research topic in the field of information technology and has attractive promising opportunities for researchers. Recognition of this field and subsequently, search for finding answers to numerous questions in this emerged context and contribute to fulfilling the promise regions in the process mining, are the motivations to select such subject [1]. In the context of process discovery, usage of heuristic and meta-heuristic algorithms exists in the researches. However, the usage of meta-heuristics methods in process discovery is rare and not common. Indeed, in this area, the just genetic algorithm (GA) and the combination of genetic and simulated annealing (SA) have been used, while the other heuristic and innovative methods have been used widely. The reason for choosing meta-heuristics methods is that they don't have any restriction to discover any structure [2]. Alves in his Ph.D. thesis has discussed this subject in detail.

In compare with other methods, meta-heuristics are theoretically superior in the process discovery [2] and also these methods are faced with some challenges. Additionally, it should be noted that the other methods are in progress as well as meta-heuristics. To overcome the challenges ahead, meta-heuristics require further and effective researches to use their maximum potential in the process discovery.

In this paper, a cloud theory-based simulated annealing is presented in which a casual matrix and a new function are used to display and evaluate the answer. In this paper, by designing two different tests with using statistical assumption tests and the same event log, the performance of this algorithm will be investigated.

2. Background and literature review

Many types of research have been done in the context of model discovery from event log which the first one was conducted in 1995 by J.E.Cook [3]. J.E.Cook proposed three methods in the context of model discovery. Aalst and Weijters in 2004 introduced the idea of process mining [4]. In their article, in addition to the process discovery, two other processes have been proposed as activities related to process mining conformance checking and enhancement.

Aalst et al. presented alpha algorithms in 2004 [5]. This algorithm is one of the first and most important algorithms for process discovery but discovered model by this algorithm has some problems. One of the problems of this algorithm is non-recognition of the distinction between models that differ in structure but same in recorded behavior [5]. Alves et al. in 2004 inspected one of the problems of the alpha algorithm, "The problem of the short loop that means the rings with one and two lengths" [6]. In this paper, with some changes in the basic algorithm, this problem is solved and Alpha+ algorithm is presented. Wen et al. in 2007 examined another problem of the alpha algorithm,

which was the problem of non-local dependencies. It means that the model created by an algorithm, show other behaviors in addition to the behavior observed in the event log [7].
Alves et al. in 2007 suggested a method for using meta-heuristics in this context. They presented an approach for applying the genetic algorithm. Their approach was based on the discovery of Petri nets. Petri nets are an approach to present the process model [8]. Bratosin et al. in 2010 improved this approach and tried to decrease the time taken in models evaluation stage by a sampling of event log [9]. In the same year and in another article, they tried to reduce the running time of algorithm by using a distributed approach [9]. Tsai et al. in 2010 added time perspective exploring to the genetic algorithm by the use of available data on the events time in the event log and incorporated obtained data by a mechanism in process model [10].
Buijs et al. in 2012 changed the process model language of Petri nets to the process tree. Search operators were also naturally changed. Moreover, the evaluation function was also changed [11]. Alizadeh and Khezerou in 2014 imposed changes on function evaluation of process tree and solved it by a combination of genetic algorithm and simulated annealing, till the impressive results were achieved and it shows the top functionality of metaheuristics [12]. Other respected works, focused on diverse computational aspects on different models including data mining with various algorithms can be mentioned [13-14-15-16].
Table I shows event log includes eight activities used in current study.

TABLE I: Event log includes eight activities used in this paper

Trace	#
acdeh	455
abdeg	191
adceh	177
abdeh	144
acdeg	111
adceg	82
adbeh	56
acdefdbeh	47
adbeg	38
acdefbdeh	33
acdefbdeg	14
acdefdbeg	11
adcefcdeh	9
adcefdbeh	8
adcefbdeg	5
acdefbdefdbeg	3
adcefdbeg	2
adcefbdefbdeg	2
adcefdbefbdeh	1
adbefbdefdbeg	1
adcefdbefcdefdbeg	1

3. The proposed method

In general, SA is a random search to find the optimal solution in NP-hard problems. These approaches are based on physical concepts of the gradual cooling process after increasing temperature to reach a high value and finally, reach to a state of a minimum potential energy. This improving mechanism starts with a primary solution and primary temperature (T0). T0 is decreased according to the cooling schedule function. Reaching thermal equilibrium is time consuming and in this interval, another solution is found in the neighbourhood of the previous solution. If the value of objective function is less than the previous value in minimization problems, the new solution will be accepted, otherwise will be accepted with probability $P=e^{\frac{-\Delta}{T}}$, where $\Delta=\frac{Fn-Fs}{Fs}$ and T is the current temperature. This process is continued until the desired state of the algorithm is reached [17].

The initial information for starting the simulated annealing algorithm is as follows:

The initial temperature (T0) for which the algorithm starts is to be set. In fact, at the beginning of the start of the algorithm, the Boltzmann probability function yields 1. In other words, at the beginning of the algorithm's start point, the results of worse neighbour be accepted with the probability of one, and they go back to zero. Figure 1 illustrates this issue.

The temperature reduction rate (α) is considered in equation 1:

$$\boldsymbol{T_h = \alpha \times T_{h-1}; h > 2, 0 < \alpha < 1} \quad (1)$$

Where $\boldsymbol{T_h}$ is the temperature in generation h and $\boldsymbol{\alpha}$ represents the rate of cooling scheduling.

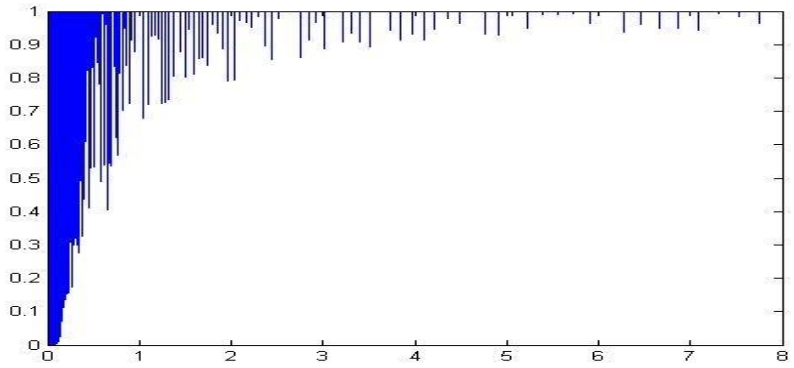

Fig. 1. Regulated Temperature graphof each generation in terms of the Boltzmann function

3.1. Internal Representation

Internal representation used in all algorithms in this paper is as a causal matrix, according to Alves and colleagues in their paper. Figure 2 shows an example of the causal matrix used in this paper.

				INPUT					
	true	*A*	*A*	*A*	*D*	*D*	$E \wedge F$	$B \vee C \vee G$	
→	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*OUTPUT*
A	0	1	1	1	0	0	0	0	$B \vee C \vee D$
B	0	0	0	0	0	0	0	1	*H*
C	0	0	0	0	0	0	0	1	*H*
D	0	0	0	0	1	1	0	0	$E \wedge F$
E	0	0	0	0	0	0	1	0	*G*
F	0	0	0	0	0	0	1	0	*G*
G	0	0	0	0	0	0	0	1	*H*
H	0	0	0	0	0	0	0	0	*true*

Fig. 2: A causal matrix is used for the internal representation of an individual.

3.2. Fitness Function

Fitness function provides an indicator of individual performance in problem space. For example, on a problem that the objective is minimization, the most appropriate individual in objective function has the lowest quantity. These raw data are commonly used as an intermediate step in determining the relative performance of individuals in the genetic algorithm.

Z1, as shown in equation 2 is the ability of a model to generate the traces observed in the event log [18].

Let L be an event log and PM be a process model [17]. Then:

$$Z1 = \frac{allproperlycomplete \log trace(PM, L)}{numtrace \log(L)} \quad (2)$$

Z2, as shown in equation 3 is the ability of a model to generate behavior observed in the log [18].

$$Z2 = \frac{allParseActivity(PM, L)}{allActivityLog(L)} \quad (3)$$

Let L be an event log and PM be a process model, the function parse tries to tracing the input trace using the model. It returns the number of activities that were successfully parsed using the model.

The function Z3, as shown in equation 4 returns percentage of loops that have been generated by the model.

Let PM be a process model, then:

$$Z3 = 1 - \frac{numLoopTrace(PM)}{allTrace(PM)} \quad (4)$$

In the fitness function, a linear combination of the three criteria of F1, F2, and F3 was put maximum and in the second fitness function, the additional traces produced by the process model are minimal. In fact, in this way, model preciseness would increase.

$$\boldsymbol{Fitness = 0.5Z1 + 0.4Z2 + 0.1Z3} \quad (5)$$

The stop condition, considering cooling pretension and the Boltzmann probability function, is considered to be 400 generations of algorithm repetition.

3.3. Cloud theory-based simulated annealing algorithm

There is a refined version of simulated annealing algorithm, called Cloud theory-based simulated annealing (CSA) algorithm, which has two important characteristics:

- The temperature of each stage is a random variable and the average become low at each stage.
- The average temperature of each stage is also a random variable (the origin of the cloud), where the variance of this random variable gradually decreases.

A cloud is a collection of random numbers that follow the rules of the normal distribution function defined by the following three attributes:

1) Cloud center 2) Cloud coverage range 3) Degree of dispersion of cloud droplets relative to each other.

Selecting appropriate initial temperature and operator for neighboring search in this algorithm is similar to the simulated annealing algorithm [17].

3.4. Operator of SA and CSA for neighborhood search

The operator used in this algorithm is the swap operator; in this way, two elements are randomly selected among top triangle elements of the causal matrix and the substitution operation is performed.

4. Experiments

In this experiment, Single target Algorithms of SA and CSA have been run ten times with respect to the operators and the parameters settings of the preceding sections and using an event log; then the behavior of the algorithm has been investigated.

4.1. First experiment

In this experiment, the initial answer has a significant effect on the final result of the algorithms. For this purpose, each of the algorithms was run with different initial answers. In fact, it was investigated according to the statistical methods and the algorithm behavior hypothesis tests are presented in Table II.

Given that the assumption H0 is not rejected, it can be concluded that the initial answer in the SA and CSA algorithms has no significant effect and with any initial answer, the algorithms are able to be the best answer.

TABLE II: Statistical hypothesis for experiment number one

	Fixed initial answer		different initial answer		Results
SA	R1,R2…R10	$\mu_1=\sum_1^{10} R_i/10$	R1,R2…R10	$\mu_2=\sum_1^{10} R_i/10$	H0 is not Rejected
CSA	R1,R2…R10	$\mu_1=\sum_1^{10} R_i/10$	R1,R2…R10	$\mu_2=\sum_1^{10} R_i/10$	H0 is not Rejected

4.2. Second experiment

In this experiment, to make sure whether there is a significant difference between the algorithms in terms of performance, we examine the behavior of algorithms in four equal time intervals. For this purpose, the algorithms are run 10 times, and the behaviors of the algorithms are investigated in a quarter of the time (¼) as presented in Table III.

TABLE III: Statistical hypothesis for experiment number two

	SA	CSA	Results
0.25	$\mu_1=\sum_1^{10} R_i/10$	$\mu_2=\sum_1^{10} R_i/10$	H0 is not Rejected
0.5	$\mu_1=\sum_1^{10} R_i/10$	$\mu_2=\sum_1^{10} R_i/10$	H0 is not Rejected
0.75	$\mu_1=\sum_1^{10} R_i/10$	$\mu_2=\sum_1^{10} R_i/10$	H0 is not Rejected
1	$\mu_1=\sum_1^{10} R_i/10$	$\mu_2=\sum_1^{10} R_i/10$	**H0 is Rejected**

According to the results obtained from the table above, H0 assumption has been rejected at the last time quarter (¼) and the algorithm is better that has a superior average according to the maximization of the objective function. The conclusion drawn from this experiment demonstrates the performance superiority of Cloud theory-based simulated annealing algorithm to simulated annealing algorithm. Figure 3 clearly shows the performance superiority of Cloud theory-based simulated annealing algorithm.

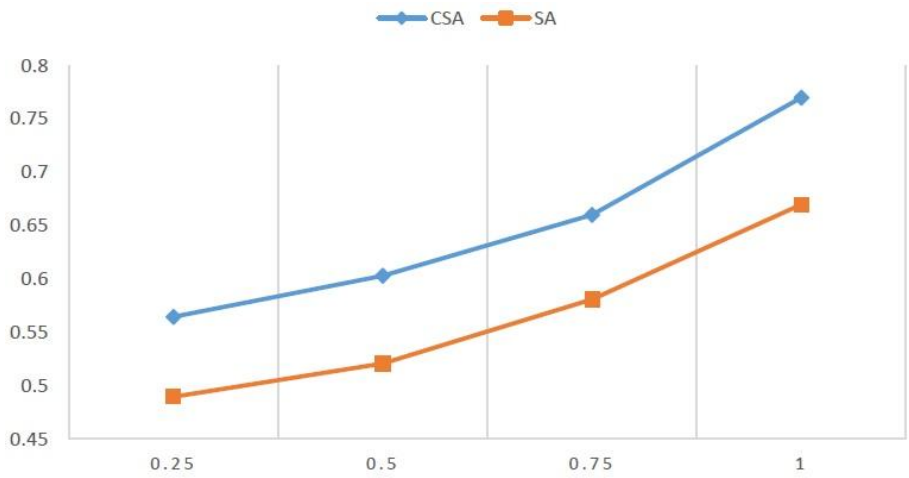

Fig. 3: Comparison of performance between two algorithms.

5. Conclusion

This article reviews the evaluation criteria of process model discovery and proposed a Cloud theory-based simulated annealing. The performance of CSA algorithm is compared with SA considering one event log. Two experiments are developed to investigate the performance of CSA. By using statistical hypothesis tests, it was found that algorithms have significant differences in their performance and finally CSA algorithm offer a better performance.

6. References

[1] V. der AalstWMP, Process Mining: Data Science in Action. Springer Verlag Berlin Heidelberg, 2016, no. 467.

[2] A. de MedeirosAK, "Genetic process mining," Ph.D. dissertation, Eindhoven University of Technology, 2006.

[3] J. Cook and A. Wolf., "Automating process discovery through eventdata analysis," in Proceedings of the 17th international conference on Software engineering, 1995, p. 7382.

[4] V. der AalstWMP and A. Weijters, "Process mining: a research agenda," Computers in Industry, vol. 53, no. 3, pp. 231 – 244, 2004.

[5] V. der AalstWMP, T. Weijters, and L. Maruster, "Workflow mining: discovering process models from event logs," IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 9, pp. 1128–1142, Sept 2004.

[6] A. de MedeirosAK, B. F. Dongen, and V. der AalstWMP, "Process mining extending the alpha algorithm to mine short loops," Technical report, WP113 Beta Paper Series Eindhoven University of Technology, 2004.

[7] L. Wen, V. der AalstWMP, J. Wang, and J. Sun, "Mining process models with non-free-choice constructs," Data Mining and Knowledge Discovery, vol. 15, no. 2, pp. 145–180, 2007.

[8] A. de MedeirosAK, A. J. M. M. Weijters, and V. der Aalst WMP, "Genetic process mining: an experimental evaluation," Data Mining and Knowledge Discovery, vol. 14, no. 2, pp. 245–304, 2007.

[9] C. Bratosin, N. Sidorova, and V. der AalstWMP, "Distributed genetic process mining," in IEEE Congress on Evolutionary Computation, July 2010, pp. 1–8.

[10] C.-Y. Tsai, H. Jen, and Y.-C. Chen, "Time-interval process model discovery and validation — agenetic process mining approach," Applied Intelligence, vol. 33, no. 1, pp. 54–66, 2010.

[11] B. JCAM, B. F. van Dongen, and V. der AalstWMP, "A genetic algorithm for discovering process trees," in 2012 IEEE Congress on Evolutionary Computation, June 2012, pp. 1–8.

[12] A. Vahedian Khezerlou and S. Alizadeh, "A new model for discovering process trees from event logs," Applied Intelligence, vol. 41, no. 3, pp. 725–735, 2014.

[13] M. Mirmozaffari, A. Alinezhad, and A. Gilanpour, "Data Mining Classification Algorithms for Heart Disease Prediction," Int'l Journal of Computing, Communications & Instrumentation Engg (IJCCIE), ISSN 2349-1469 EISSN 2349-1477, Vol.4, Issue1, Jan 2017.

[14] M. Mirmozaffari, A. Alinezhad, and A. Gilanpour, "Heart Disease Prediction with Data Mining Clustering Algorithms," Int'l Journal of Computing, Communications & Instrumentation Engg (IJCCIE), ISSN 2349-1469 EISSN 2349-1477, Vol.4, Issue1, Jan 2017.

[15] M. Mirmozaffari, A. Alinezhad, and A. Gilanpour, "Data Mining Apriori Algorithm for Heart Disease Prediction," Int'l Journal of Computing, Communications & Instrumentation Engg (IJCCIE), ISSN 2349-1469 EISSN 2349-1477, Vol.4, Issue1, Jan 2017.

[16] M. Mirmozaffari, A. Alinezhad. "Ranking of Heart Hospitals Using Cross-Efficiency and Two-stage DEA," 7th International Conference on Computer and Knowledge Engineering (ICCKE 2017), October 26-27 2017, Ferdowsi University of Mashhad, 978-1-5386-0804-3/17/$31.00 ©2017 IEEE.

[17]E. Torabzadeh, M. Zandieh, "Cloud theory-based simulated annealing approach for scheduling in the two-stage assembly flowshop," Advances in Engineering Software, vol. 41, no. 10, pp. 1238 – 1243, 2010.

[18] A. de MedeirosAK, A. Weijters, and V. der AalstWMP, "Using genetic algorithms to mine process models: representation, operators, and results," Eindhoven: Technische Universiteit Eindhoven, 2004.