http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 10 (6): 1240-1245, 2011
ISSN 1812-5638 / DOI: 10.3923/itj.2011.1240.1245

An Efficient Process Mining Method Based on Discrete Particle Swarm Optimization

Xianwen Fang, Xin Gao, Zhixiang Yin and Qianjin Zhao
School of Science, Anhui University of Science and Technology, Huainan Anhui, 232001, China

Abstract: Process mining is to extract business process models from event logs, the mining process is an important learning task. However, the discovery of these processes poses many challenges, including noise, non-local, non-free choice constructs and so on. In the study, we give out the definition of the behavior redundancy degree which is benefit to analyze the behavior conformance. Then, in order to build the optimal the process model, a process mining method based on Discrete Particle Swarm Optimization (DPSO) is presented. The method can take into account the basic Petri net structure and the metrics of behavior conformance and avoid the blindness of building process model. Finally, a DPSO process mining plug-in is developed and a number of event log is tested in the DPSO mining plug-in based on PROM platform. Theoretical analysis and experimental results show that DPSO-based mining method has better behavior fitness and behavior appropriateness in business process mining.

Key words: Process mining, discrete particle swarm optimization, petri net, behavior conformance

INTRODUCTION

The goal of process mining is to extract information about processes from transaction logs. Event logs are used as the starting point for mining. It is possible to record events such that (1) each event refers to an activity (i.e., a well-defined step in the process), (2) each event refers to a case (i.e., a process instance), (3) each event can have a performer also referred to as originator (the person executing or initiating the activity) and (4) events have a timestamp and are totally ordered. They can be looked at from different perspectives: (1) the process perspective, (2) the organizational perspective and (3) the case perspective. The process perspective focuses on the control flow, i.e., the ordering of activities (Fang *et al*., 2010). The goal of mining in this perspective is to find a good characterization of all possible paths, e.g., expressed in terms of a Petri net, possibly exploiting the concurrence and choice that are flattened in the paths.

Within the research domain of process mining, process discovery aims at constructing a process model as an abstract representation of an event log. The goal is to build a model (e.g., a Petri net) that provides insight into the behavior captured in the log (Van Dongen *et al*., 2009).

There are several process mining methods, one is the abstraction-based mining algorithms, the algorithms construct a net based on an abstraction of the log and most of them are derived from the α-algorithm (Van der Aalst *et al*., 2003) because the mining procedures of these algorithms are very similar, called as α-series algorithms. This mining procedure consists of three phases: the abstraction phase, the induction phase and the construction phase. In the abstraction phase, any event in each event trace in the event log is scanned and basic ordering relations between tasks are recorded. These relations contain information about the number of direct successions between tasks. In the induction phase, advanced ordering elations are induced from the basic ones and new tasks (e.g., invisible tasks and duplicate tasks) may be created from scratch. Advanced ordering relations for example show whether tasks are causally dependent or in parallel. In the construction phase, the final model is constructed from the advanced ordering relations according to some heuristic rules. But, α-series algorithms only can mine the WF-net which is a safe net and there is still not a single abstraction-based algorithm that can handle all the special constructs in a sound SWF-net.

The theory of regions can be used to transform a state-based model or a set of words into a Petri net that exactly mimics the behavior given as input. Recently several studies appeared on the application of the theory of regions for process discovery. Two types of region theory can be distinguished, namely state-based region theory and language-based region theory (Carmona *et al*., 2010). The state-based theory of regions focuses on the synthesis of Petri nets from state-based models, where the state space of the Petri net is bisimilar to the given state-based model. The language-based region theory,

Corresponding Author: Xianwen Fang, School of Science, Anhui University of Science and Technology, Huainan Anhui, 232001, China

considers a language over a finite alphabet as a behavioral specification. Using the notion of regions, a Petri net is constructed, such that all words in the language are firing sequences in that Petri net. The region-based process mining algorithms can mine the no-safe net, the algorithms obtain the places based on the minimal regions, then construct the Petri net model by analyzing the transitions. But the constructing process only considers the direct causal dependency, some indirect causal dependency relationships are not taken into account.

The genetic process mining algorithms (Alves Medeiros *et al*., 2007) can deal with noise and can be used to express the main behavior (i.e., not all the details and exceptions) registered in an event log. It supports the mining of all common constructs in process models (i.e., sequence, choice, parallelism, loops, invisible tasks and some kinds of non-free-choice), except for duplicate tasks. The genetic process mining algorithm has two main steps. In the first step, a dependency graph is built. In a second step, the semantics of the split/join points in the dependency graph are set. But the mining result is main behavior model, so some low occur frequency behavior may be not mined.

BASIC CONCEPTION

Petri net is a formal event-based modeling tool, its firing sequences can correspond well to the traces of business process, it is convenient to describe some complex system with concurrent, choice, synchronization structure and having rich behavior analysis tools and rigorous mathematical analysis methods, so Petri net have a greater advantage in dynamic building business processes. About the basic concept, structure and behavior properties of Petri nets may refer to the in the literature (Murata, 1989).

In the event log of business process, the log can be extracted through a number of tools, such as log parser plug-in of Prom, the event log can handled into. mxml file, get the traces of event log, that is case traces. We take the case traces as the corresponding legal sequences of Petri net and analyze the behavior dependent relationships between the different tasks (or transition) (Fang *et al*., 2009), in order to determine the corresponding Petri net structure of traces, the objective is to obtain the optimal business process model by the metrics methods of behavior conformance.

Behavior dependent relationship: For convenience, in this study, the traces of event log are corresponding to the legal sequences of Petri net model and taking the properties of Petri nets into account.

Definition 1 (Lijie *et al*., 2007) (Basic order relationship): Let W is a event log, N = (P, T; F) is the building P etri net of business process, here, W = L (N), a, b ∈ T:

(1) $a >_w b$ iff $\exists \sigma = t_1 t_2 t_3 \cdots t_n \in W, i \in \{1, \cdots, n-1\} : t_i = a \wedge t_{i+1} = b;$

(2) $a\Delta_w b$ iff $\exists \sigma = t_1 t_2 t_3 \cdots t_n \in W, i \in \{1, \cdots, n-2\} : t_i = t_{i+2} = a \wedge t_{i+1} = b;$

(3) $a \diamondsuit_w b$ iff $a\Delta_w \wedge v\Delta_w a$

(4) $a \rightarrow_w b$ iff $a >_w b \wedge (\neg (b >_w a) \vee a \diamondsuit_w b)$

(5) $a \#_w b$ iff $\neg (a >_w b) \wedge \neg (b >_w q)$

(6) $a \|_w b$ iff $a >_w b \wedge b >_w a \wedge \neg (b >_w (a \diamondsuit b)$

As can be seen from the definition 1, $>_w$ and Δ_w are the basic order relationship, the former represents t the two tasks occur one after another, the latter represents two tasks can generate a specific piece of the cycle track. These can be used to difference the relationships of two tasks. (3) (4) (5) (6) corresponding to cycle, sequence, select and concurrency relations, respectively.

Definition 2 (Lijie *et al*., 2007) (Direct dependent relationship): N = (P, T; F) is the building Petri net of business process, for arbitrary a, b ∈ T there exists direct dependent relationship between a and b if and only if:

(1) $a^{\bullet} \cap {}^{\bullet} b \neq \Phi$

(2) There exists reachability mark s ∈ [N, [I]> to make (N, s) [a> and (N, s-• a + a•) [b>

Definition 3 (Lijie *et al*., 2007) (Indirect dependent relationship) N = (P, T; F) is the building Petri net of business process, for arbitrary a, b ∈ T if there exists indirect dependent relationship between a and b iff:

(1) $a^{\bullet} \cap {}^{\bullet} \neq b\ \Phi$ and arbitrary p ∈ a• ∩• b, p is not the implicit place
(2) There do not exist reachability mark s ∈ [N, [I]>, to make (N, s) [a> and (N, s-• a + •) [b>;
(3) There exists reachability mark s ∈ [N, {i}>, to make (N, s) [a> and exists reachability mark s' ∈ [N, s-• a +• >,to make (N, s') [b>

The metrics of behavior conformance: Behavior conformance (Van der Aalst Ouyang *et al*., 2008) main analyzes the conformance between the models with the log, some indexes are proposed in literature. Aimed to the learning process, we propose the behavior redundancy degree.

Definition 4 (Alves de Mederios *et al*., 2008) (Behavioral precision and recall): Let (N_1, M_1) and (N_2, M_2) be marked Petri nets and let $L \in B(T^*)$ be a multi-set over T:

$$\text{precision } ((N_1,M_1), (N_2,M_2),L) = \left(\sum_{\sigma \in L} \frac{L(\sigma)}{|\sigma|} \left(\sum_{i=0}^{|\sigma|-1} \frac{|\text{enabled } (N_1,M_1,\text{hd } (\sigma,i)) \cap \text{enabled } (N_2,M_2),\text{hd } (\sigma,i)|}{|\text{enabled } (N_2,M_2,\text{hd } (\sigma,i)|}\right)\right)/|L| \quad (1)$$

$$\text{recall } ((N_1,M_1), (N_2,M_2),L) = \left(\sum_{\sigma \in L} \frac{L(\sigma)}{|\sigma|} \left(\sum_{i=0}^{|\sigma|-1} \frac{|\text{enabled } (N_1,M_1,\text{hd } (\sigma,i)) \cap \text{enabled } (N_2,M_2),\text{hd } (\sigma,i)|}{|\text{enabled}(N_1,M_1,\text{hd } (\sigma,i)|}\right)\right)/|L| \quad (2)$$

where, hd (r, k) = $<a_1, a_2, ...a_q>$, i.e., the sequence of just the first k elements.

Definition 5 (Redundancy degree): Assume the firing probability of all transition sequences is the same, the largest frequency of each sequence is the MAX, MAX = max $\{L(\sigma) | \sigma \in (N, M)$ and $\sigma \in L\}$, then the redundancy degree of Petri net based on the event log is

$$\text{RED } ((N,M),L) = \left(\sum_{\sigma \in L} \frac{\text{MAX} - L(\sigma)}{|\sigma|} \left|\left\{i \in \{0,|\sigma|-1\} | \sigma(i+1) \notin \text{enabled } (N,M,\text{hd } (\sigma,i))\right\}\right|\right) \Big/ |L| \quad (3)$$

where, hd (r, k) = $<a_1, a_2, ..., a_k>$, i.e., the sequence of just the first k elements.

THE PROCESS MINING METHOD BASED ON THE DISCRETE PARTICLE SWARM OPTIMIZATION

The basic knowledge of DPSO: Particle Swarm Optimization (PSO), as a kind of adaptive evolutionary computation technique. Its optimization idea originated from the regular enlightenment from cluster behavior of birds and fishes and the natural selection mechanism of evolutionary algorithm is replaced by organizational social behavior in PSO and the optimal solution can be searched through the cooperation of individuals (Alec *et al*., 2007). In PSO, population is generated randomly and every particle in population is given a random velocity which can be adjusted dynamically according to the experiences of itself and partners during flying. PSO algorithm is similar to other evolutionary algorithms in some aspects, for example, individual is also moved to better region according to the fitness to the environment. The difference is that the evolutionary operator has not been applied yet in PSO while every individual is looked as a particle without bulk and quality which can fly in space and whose velocity can be dynamically adjusted according to the comprehensive analysis of the flying experiences of itself and population. The ith particle is denoted as $X_i = (x_{i1}, x_{i2}... x_{id})$ and V_i $(v_{i1}, v_{i2}..... v_{id})$ is the present flying velocity of particle i and $p_i = (p_{i1}, p_{i2}...p_{id})$ is the best place that particle i has experienced, that is to say, P_i is the place with optimal fitness value and called as local best position. P_g, called as global best position, is the best position that all particles have experienced.

The standard PSO is applicable for continuous problems while process mining is a kind of discrete problem. So it has become unsuitable for mining the process model, therefore, combining with the properties of the Petri net, we propose DPSO method to solve the process mining of business process.

Definition 6 (Substitution Operation): For the task t_i $(1 \le i \le n)$ in process model, ts_{ij} is the selected trace sequence in the t-th generation and if the selected trace sequence in the (t + 1)-th generation is ts_{ik}, $ts_{ij} \rightarrow ts_{ik}$ is defined as the substitution from the t-th to the (t + 1)-th generation, in which ts_{ij} is the substituted trace sequence and ts_{ik} is the substituting trace sequence. It is denoted as:

$$\text{substitute } (t_i^{t+1}) = ts_{ik}$$

Specially, there is no operation if ts_{ij} is the same as ts_{ik}, called as NOP operation. For example, for the particle P_a and P_b, if the values in the d-th dimension are ts_j and ts_k, respectively, P_{ad}-P_{bd} means $t_{sk} \rightarrow t_{sj}$.

So, we use the substitution operation, the velocity and place is represented as follows:

$$v_{id}^{t+1} = \omega v_{id}^t \oplus c_1 r_{1d}(P_{id} - x_{id}) \oplus c_2 r_{2d}(P_{gd} - x_{id}) \quad (4)$$

$$x_{id}^{t+1} = \text{substitute } (t_{id}^{t+1}) \quad (5)$$

where, ω is the inertia weight which can make particle keep movement inertia and can expand the searching space so as to have ability to explore new position. The larger ω can make particle possess better global search ability while less ω makes particle possess better local search ability and experimental results show that algorithm has better comprehensive performance when ω takes value from [0.8,1.4] (Alec *et al*., 2007).

c_1 and c_2 are accelerating constants and c_1, also named "cognitive coefficient", reflects the effect of local best position on particle flying velocity and c_2 also named "social learning coefficient" reflects the effect of global

best position on particle flying velocity, $r_1 \sim U(0, 1)$, $r_2 \sim U(0, 1)$, are two random variables independent each other.

The modeling and optimization of the process mining based on DPSO: Though analyzing business processes mining, its traits are accord with the features of DPSO learning process. Because we can take the case traces as the observed sequences of business processes, the frequency of every case can be looked as the statistical corresponding relationships between the observed sequences and the state, the process meets the stochastic process of DPSO, using the DPSO thought to build the business process model is feasible. And we present effective metrics methods of behavior conformance, for example the behavior redundancy degree; the DPSO process mining should have better behavior fitness and behavior appropriateness.

The DPSO-based process mining algorithm is as follows:

Step 1: Log sequence must be pretreated first; mostly the same sequences will be merged. Then analyzing the direct dependent relationships between all tasks (or transition)

Step 2: Building the initial model λ_0, according to the pretreated sequences, using the direct dependent relationships to build the initial Petri net model. In building the model, we do not consider the metrics of behavior conformance and other factors, only consider the behavior dependent relationships of tasks (transition), the built model is called as λ_0

Step 3: Checking whether the initial model λ_0 accepts the sequence, if some sequences can be accepted, then learn the typical behavior sequences based on the frequency of sequences (i.e., the behavior sequences of large frequency should be accepted), obtaining the new model by learning process

Step 4: According to the definition of structure fitness, behavior precision and behavior recall, computing the metrics value of model λ_0 and model λ based on the log sequences, if (fitness $(\lambda, L) \leq$ fitness $(\lambda_0 L)$ and precision $(\lambda, \lambda_0, L) \geq \omega 1$) or (fitness $(\lambda, L) \geq$ fitness (λ_0, L) and recall $(\lambda, \lambda_0, L) \geq \omega 1$), then return to step (5); else, the initial model λ_0 keeps unchanging, repeat the step (3), continue to learn and select the next typical behavior sequences, the selecting consider the different structures with the same sequences, the basic Petri net structure with the same firing sequences can be referenced

Step 5: If recall $(\lambda, \lambda_0, L) \geq$ precesion (λ, λ_0, L) then λ_0 return to the step 6; else, computing the redundancy degree RED (λ_0) and RED $(\lambda)$of the model λ_0 and the model λ, respectively, if:

$$\frac{\text{fitness}(\lambda, L)}{\text{RED}(\lambda)} \geq \frac{\text{fitness}(\lambda_0, L)}{\text{RED}(\lambda_0)}$$

then $\lambda_0 = \lambda$, return to the step 6; else, determine whether the model λ_0 has the indirect dependent relationship of tasks (transitions), if there exist the indirect dependent relationship, the learning process by the basic structure of indirect dependent relationship and return to the step 4

Step 6: If:

$$\text{fitness}(\lambda_0, L) \geq \omega 3 \;\&\; \& \frac{\text{fitness}(\lambda_0, L)}{\text{RED}(\lambda)} \geq \omega$$

then the learning process end and the maximum possibility is λ; else, return to the step 4 and continue to learn, until the learning process end

Step 7: Output the model λ

In the training process, the size of each ω values can be set according to the size of the sequences and we set the three ω values to equal to 0.8 in the experiment. Viewed from the algorithm, if the behavior recall value of the new model λ is greater than or equal to the behavior precision value, this indicates the coverage degree of the new model λ greater than the original model λ_0. So we can use the new mode λ to replace the original model λ_0. In this case, we can use the "if the modeling is end?" to determine whether the new model λ is the maximum possibility model. If the behavior recall value is less than the behavior precision value of the new model λ, then the new model need to learn according the circumstances of the original model λ_0, it is necessary to determine the redundancy degree of the new model λ and the original model λ_0 and compare the value of fitness/RED, if the fitness/RED value of new model λ is larger, indicates that the redundancy of the new model is smaller, so the new model λ can be used to replace the original model λ_0, then further to continue to learn.

EXPERIMENT SIMULATION

Based on the proposed DPSO process mining methods, we develop the DPSO mining plug-ins on the ProM framework. The ProM framework (Alves de Medeiros *et al*., 2008) is an open-source tool and it can be downloaded at www. processmining. org., specially tailored to support the development of process

mining plug-ins. In ProM, plug-ins can be categorized. Firstly, using the DPSO mining plug-ins, according to the case traces, the relationship of tasks (transition) can by analyzed and the simple Petri net model can be built by the event log. Secondly, through computing and comparing the metrics of behavior conformance, the simple Petri net model is optimized by learning and the learning process is according to the basic structure of the same event traces. Finally, using the checking plug-in to check the behavior fitness and behavior appropriateness between the obtained model and the event log and obtain the optimization mined model.

Here, we give out several snapshots of the DPSO mining process, the snapshot of the conformance analyzing between the model and log is showed in Fig. 1, the snapshot of DPSO learning result is showed in Fig. 2.

We compare the DPSO process mining methods with the genetic process mining methods (Alves de Medeiros *et al*., 2008). Experiment environment: CPU is Intel dual 1.60 GHz, Memory is 2.00 GB and operation system is Windows XP. DPSO mining plus-ins is developed based on the ProM with version 4.2.

For the same event logs from the large Benchmarks (Van der Aalst *et al*., 2008), using he DPSO process

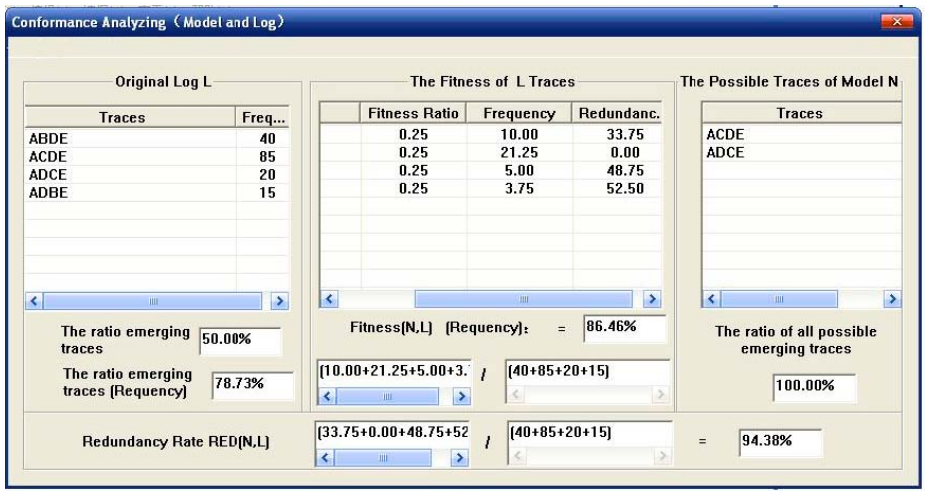

Fig. 1: The snaphhot of the conformance analyzing between the model and log

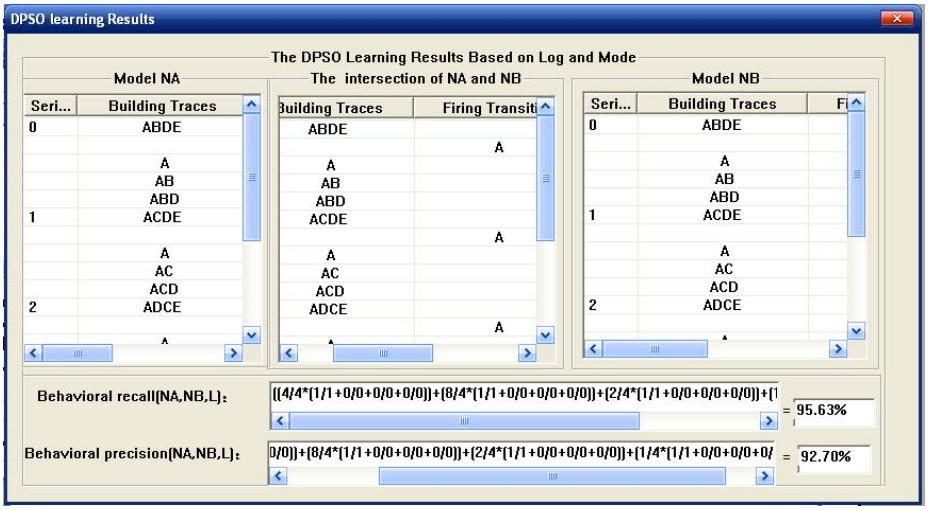

Fig. 2: The snapshot of DPSO learning result

Table 1: The results of two methods based on large benchmarks

		a22f0n00-1		a22f0n00-2		a22f0n00-3		a22f0n00-4		a22f0n00-5	
	Log	Case	[S]	Case	[S]	Case	[S]	Case	[S]	Case	[S]
Methods	Index	100	322	300	993	500	1170	700	585	900	619
DPSO methods	Cost	5.34sec		7.449sec		10.05sec		13.96sec		18.12sec	
	Beh-Fit.	1.00		0.993		0.956		0.949		0.936	
	Beh-App	0.992		0.988		0.995		0.987		0.992	
Genetic methods	Cost	6.81sec		9.13sec		13.29sec		16.95sec		22.17sec	
	Beh-Fit.	0.896		0.893		0.893		0.893		0.89	
	Beh-App	0.983		0.978		0.988		0.985		0.989	

mining methods and the genetic process mining methods respectively, we compare the three indexes including time-cost, behavior fitness and behavior appropriateness, in order to analyze the relationships between the mined models with the event log by two process mining methods.

In Table 1, with the case increase, the time-cost of two methods are both increasing quickly but the time-cost of the DPSO mining methods is less than the time-cost of the genetic mining methods. The reason is that the DPSO methods can accelerate the learning process. The fitness and appropriateness of two methods are irrelevant with the case number but our methods are better than the genetic process mining methods.

CONCLUSIONS

The study presents a mining and analyzing method of business process based on DPSO. Firstly, aimed to the compare between model with log, we give out the metrics of behavior conformance (main index is behavior redundancy degree). Then, in order to build the optimal the process model, we present the DPSO process mining method. The method can take into account the basic Petri net structure and the metrics of behavior conformance and avoid the blindness of building process model. Finally, using methods above mentioned, we develop the DPSO mining plug-ins based on the Prom version 4.2 and compare the DPSO methods with genetic process mining methods. Theoretical analysis and experimental results indicate that the DPSO mining methods are better than the methods of genetic process mining methods.

In the future, we would like to study the process mining methods with the inducement information. Moreover, it is also one of our future works to study the intelligence mining methods of complex business process.

ACKNOWLEDGMENTS

We would like to thank the support of the National Natural Science Foundation of China under Grant No. 60873144, No. 60973050 and No. 61073102, the National High-Tech Research and Development Plan of China under Grant No. 2009AA01Z401, the Natural Science Foundation of Educational Government of Anhui Province of China (KJ2009A50,KJ2010B310, KJ2011A086).

REFERENCES

Alec, B., V. Jonathan and A. Chukwudi, 2007. A review of particle swarm optimization. Part I: Background and development. Nat. Comput. Int. J., 6: 467-484.

Alves Medeiros, A.K.A., A.J.M.M. Weijters and W.M.P. van der Aalst, 2007. Genetic process mining: An experimental evaluation. Data Mining Knowledge Discovery, 14: 245-304.

Alves de Medeiros A.K., W.M.P. Van and A.J.M.M. Weijters, 2008. Quantifying process equivalence based on observed behavior. Data Knowledge Eng., 64: 55-74.

Carmona, J., J. Cortadella and M. Kishinevsky, 2010. New region-based algorithms for deriving bounded petri nets. IEEE Trans. Comput., 59: 371-384.

Fang, X., C. Jiang and X. Fan, 2009. Independent global constraints-aware web service composition optimization. Inform. Technol. J., 8: 181-187.

Fang, X., C. Jiang and X. Fan, 2010. Behavior-aware trustworthiness study of networked software. Int. J. Comput. Intell. Syst., 3: 542-552.

Lijie, W., W.M. Aalst, J. Sun and W. Jianmin, 2007. Mining process models with non-free-choice constructs. Data Mining Knowledge Discovery, 15: 145-180.

Murata, T., 1989. Petri nets - properties, analysis and applications. Proc. IEEE, 77: 541-580.

Van Dongen, B.F., A.K. Alves de Medeiros and L. Wen, 2009. Process mining: Overview and outlook of petri net discovery algorithms. LNCS, 5460: 225-242.

Van der Aalst, W.M.P. and C. Ouyang, 2008. Conformance checking of service behavior. ACM Trans. Internet Technol., 8: 1-30.

Van der Aalst, W.M.P., B.F. van Dongen, J. Herbst, L. Maruster and G. Schimm, 2003. Workflow mining: A survey of issues and approaches. Data Knowledge Eng., 47: 237-267.