

Optimizing Resource Conflicts in Workflow Management Systems Using an Innovative Algorithm

Darshana Patel¹, Mitula Pandya²

¹Pursuing M.E., Gujarat Technological University, Alpha College of Engineering & Technology, Gandhinagar, Gujarat, India

²Professor, Gujarat Technological University, Alpha College of Engineering & Technology, Gandhinagar, Gujarat, India

Abstract: Resource allocation and scheduling are fundamental issues in a Workflow Management System (WfMS). Effective resource management in WfMS should examine resource allocation together with task scheduling since these problems impose mutual constraints. Optimization of the one factor is subject to the other constraints and vice versa. Thus, an ideal algorithm should take into account not only performance metrics of the infrastructure, such as the number of resources and their utilization, but also quality criteria such as the percentage of tasks undergone violation in their temporal restrictions. In this paper, we propose an innovative algorithm which jointly optimizes the two aforementioned contradictory criteria. The algorithm, called Resource Conflicts Joint Optimization, minimizes resource conflicts subject to temporal constraints and simultaneously optimizes throughput or utilization subject to resources constraints. To achieve the optimization, the two factors are formulated in a matrix form and the optimal solution is found by applying concepts of the generalized eigenvalue analysis. A rough outline of an agent-based architecture is proposed to achieve runtime integration of our algorithm into a functional WfMS, while experimental results under different load environments and tasks assumption reveal the superiority of the proposed strategy than the other conventional approaches.

Keywords: Data mining, Process mining, WF-Net, Alpha Algorithm, Heuristic Miner Algorithm

1. Introduction

To propose an innovative algorithm which jointly optimizes the two contradictory criteria as mentioned in abstract. The algorithm will minimize resource conflicts subject to sequential constraints and simultaneously optimizes the amount of material or items passing through a system or process or utilization subject to resources constraints. For the achievement of the optimization eigenvalue analysis have been presented. It not only proved itself useful for optimization but also useful to get the optimum solution by applying concepts. The purpose of using eigenvalue analysis is mainly to search the best solution after calculation of the processes under the algorithm roof.

2. Overview

Numerous type of information systems record event data about executed tasks. Workflow mining is concerned with the origin of a graphical workflow model out of this data. Workflow mining is a highly interactive process. The mining expert repeatedly approaches the result by changing the parameter of the mining tool and verifying the mined models. The modeling of business process in a computerized manner either fully or partially is called as workflow.

Workflow is used for representing a repeatable sequence of operations of a business process. It is a activity pattern used for systematic organization of resources, investigation, operational implementation to represent real work for further assessment. Workflow is related to various fields like operations research, artificial intelligence or in some special type of industries like printing, automation of events or where processes are executed by particular rules for completing the

business process. A workflow management system is a computer software system that manages and specifies a series of tasks to generate a final outcome within an organization. Workflow Management System defines various workflows for different types of processes. Workflow management concepts and technologies have been functional in many enterprise information systems.

Many software systems ERP (Enterprise Resource Planning) systems such as SAP, PeopleSoft, and Oracle, CRM (Customer Relationship Management) software, etc. have adopted workflow technology. Even with its promise, many problems are encountered while applying workflow knowledge. One of the problems is that these systems require a workflow design, i.e., a designer need to construct a detailed model accurately unfolding the routing of work. Modeling a workflow is far from trivial: It requires understanding of the workflow language and detailed discussions with the workers and management involved. By creation of the graphical process definitions, i.e., models describing the life-cycle of a classic case (workflow instance) in segregation, one can classify these systems to maintain business processes.

3. Background Theory

3.1 Workflow: Where and How

- It supports mainly the analysis and the design of the new system. It helps to understand how people work with the new system.
- It reduces the analysis time and cost because in most of information system, log data is readily available without any additional cost.

- It provides objective information about any system under consideration because workflow logs are an impartial reflection of executed activities.
- It allows formal verification of workflow properties by simulating the process before its execution.
- It supports multiple views of the same process by synthesis algorithms and model conversions.
- It allows the involuntary derivation of special paths from acyclic normal flow of the process whenever required.
- It enacts repeated model changes on the running instance of a workflow, shielding them from undesired outcomes.

3.2 The main objectives of a Human-Centric Workflow solution are:

- What is the current Status of a business process?
- Who is the owner for a business process?
- Who need to be informed?
- What happened so far to the business process?
- Who is allowed to access and modify data?
- The Workflow Engine automatically routes open tasks to the next actor and notifies users about new tasks depending on the current process definition. The following illustration demonstrates the typical flow of a task from one user to another controlled by a Workflow Management System.

3.3 Classification in Data Mining

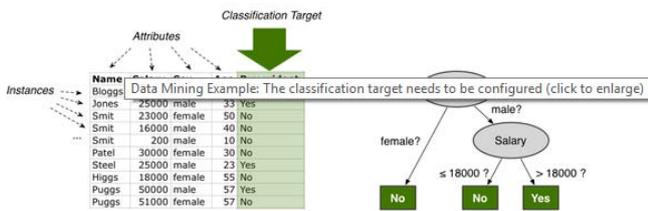


Figure 1: Classification of data and decision tree

Out of all fields of a dataset some necessary fields should be sorted out. These are the fields that allows data to be classify and a decision tree can be generated as given at the right side of the above diagram

3.4 Event Logs

The main aim behind recording an event is to find the exact process being executed at once. To fulfill the core idea behind the process mining, proper maintenance of event log should be kept.



Figure 2: Event log sample

If you look at the highlighted rows 6–9, you can see one process instance (case9705) that starts with the status Registered on 20 October 2009, moves on to at specialist and in progress, and ends with status Completed on 19 November 2009.

In process mining, looking the history of the data process is useful. This way it becomes easy to perform operations on and to decide where the data come from.

3.5 Process Model

Using process discovery, we can generate results of the selected fields. The fields should contain some weight and the connection between the processes internally can be classified based on that.

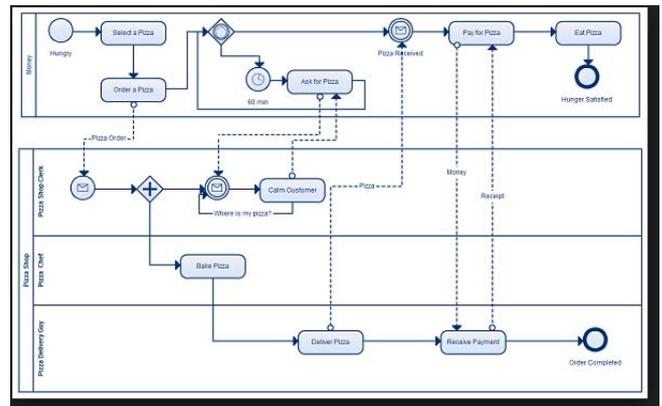


Figure 3: Process Model Sample

We can understand the process flow by seeing the process model. Generally, it is drawn differently by connecting each part of the process unit. The main idea behind the process model should be to find out important process out of all.

3.6 Algorithms for Process Mining

Various process mining algorithms have been used to solve the problem of efficient process mining.

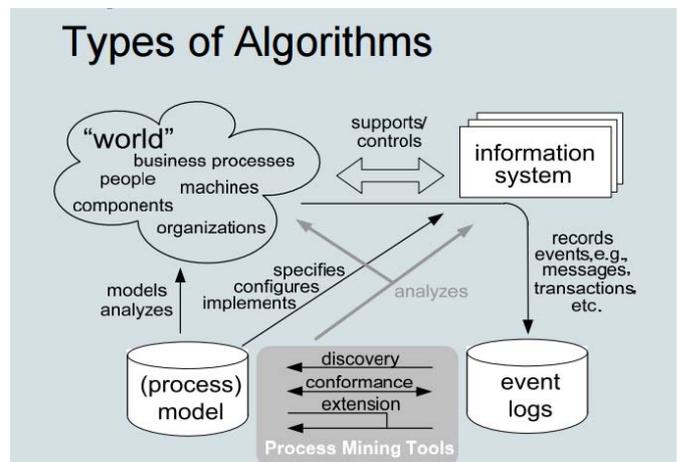


Figure 4: Process mining algorithm flow

For a process mining algorithm, generating noise free event log is very important. For such purpose there are several techniques can be applied. The event log is mainly generated

in XML format. In which <AuditTrainEntry> is a compulsory field. Than <WorkflowElementModel> and <EventType> displays the type of the task.

Based on these, the algorithm steps will be performed. The main Algorithms are alpha algorithm, heuristic miner algorithm, genetic algorithm and fuzzy miner algorithm.

• Alpha algorithm

1. Read a log
2. Get the set of tasks
3. Infer the ordering relations
4. Build the net based on inferred relations
5. Output the net

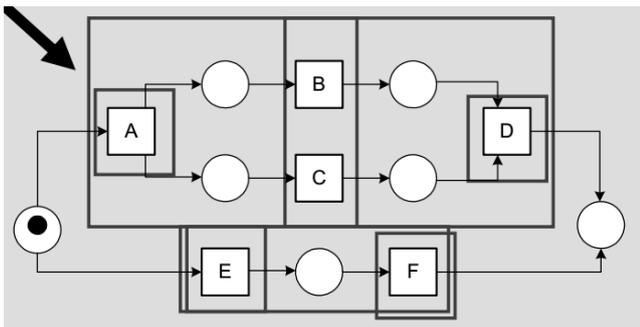


Figure 4: WorkFlow Net

• Heuristic Miner Algorithm

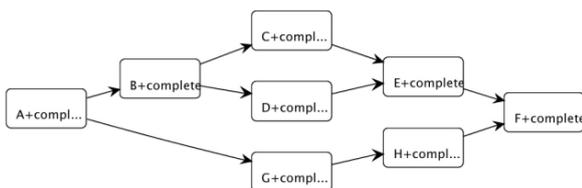
1. Read a log
2. Get the set of tasks
3. Infer the ordering relations based on their frequencies
4. Build the net based on inferred relations
5. Output the net

Core idea behind the heuristic miner algorithm is the more frequently a task A directly follows another task B, and the less frequently the opposite occurs, the higher the probability that A causally follows B!

Local approaches

- α-algorithm, Heuristics Miner
- Global approaches
- Genetic Miner and Fuzzy Miner

4. Proposed Work & Approach



The process petrinet is given in figure below, it has 12 tasks. It is represented as c-net in the following table, which is converted using DG-Graph

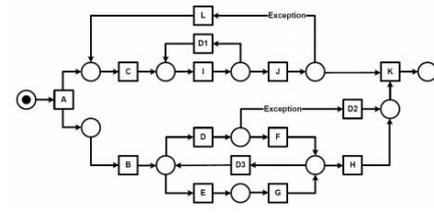


Fig. 2. The Petri net process model used as reference for generating event logs (with and without noise). The hidden tasks D1, D2, and D3 are not registered in the event-logs.

Definition 2 (Dependency Graph (DG)). If $C\text{-net} = (T, I, O)$ is a Causal net then the corresponding Dependency Graph (DG) is a relation on T ($DG \subseteq T \times T$), with

$$- DG = \{(a, b) | (a \in T \wedge b \in a \square) \vee (b \in T \wedge a \in \square b)\}$$

As an example, we show how the Petri net in Fig. 2 can be represented as a C-net (see Table 1). The Petri net in Fig. 2 has 12 tasks (A, B, \dots, L), so the corresponding task set $T = \{A, B, \dots, L\}$.

Table 1. The translation of the Petri net (Fig. 2) into a C-net.

I	ACTIVITY	O
{}	A	{{B}, {C}}
{{A}}	B	{{E}, {D}}
{{A}, {L}}	C	{{I}}
{{B}, {F}, {G}}	D	{{F}, {K}}
{{B}, {F}, {G}}	E	{{G}}
{{D}}	F	{{D}, {E}, {H}}
{{E}}	G	{{D}, {E}, {H}}
{{F}, {G}}	H	{{K}}
{{C}, {I}}	I	{{I}, {J}}
{{I}}	J	{{K}, {L}}
{{J, H}, {J, D}}	K	{}
{{J}}	L	{{C}}

Table 2. Direct successor ($a \succ_w b$ -counting) and frequency (last line) counting.

	Start	A	B	C	D	E	F	G	H	I	J	K	L	End
Start	0	1000	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	520	480	0	0	0	0	0	0	0	0	0	0
B	0	0	0	360	182	198	0	0	0	233	27	0	0	0
C	0	0	338	0	125	128	40	48	8	349	0	0	0	0
D	0	0	0	63	0	0	586	0	0	193	68	5	6	0
E	0	0	0	73	0	0	0	619	0	236	67	0	3	0
F	0	0	0	16	124	134	0	0	327	212	88	0	7	0
G	0	0	0	16	143	145	0	0	359	220	105	0	10	0
H	0	0	0	11	0	0	0	0	0	252	105	614	5	0
I	0	0	119	0	209	236	179	210	166	315	576	0	0	0
J	0	0	23	0	135	155	102	117	118	0	0	381	5	0
K	0	0	0	0	0	0	0	0	0	0	0	0	0	1000
L	0	0	0	17	3	2	1	4	9	0	0	0	0	0
End	0	0	0	0	0	0	0	0	0	0	0	0	0	0
#	1000	1000	1000	1036	921	998	908	998	987	2010	1036	1000	36	1000

Table 3. Length-two loops counting ($a \gg_w b$ -counting). The value 89 in position D,F indicates that there are 89 DFD patterns in the event log. Remark the high value between LIL-task I and many other tasks (i.e., B, D, E F, G and H). This is caused by the looping behavior of I in combination with the parallel behavior of the other mentioned tasks.

	Start	A	B	C	D	E	F	G	H	I	J	K	L	End
Start	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	89	0	0	0	0	0	0	0
E	0	0	0	0	0	0	0	104	0	0	0	0	0	0
F	0	0	0	0	110	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	133	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	19	0	40	63	59	57	97	116	0	0	0	0
J	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	0	0	0	0
End	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1. The first step of the algorithm is the construction of the set A (the set of all tasks appearing in the log).

2. Looking at the diagonal of Table 2 there is only one candidate for C1: task I is 315 times followed by itself. The value of $I \Rightarrow W I = 315 / (315 + 1) \geq \sigma_{L1L}$, resulting in $C1 = \{I, I\}$.

3. For this step of the algorithm we make use of Table 3. The table indicates that pattern DF D appears 89 times in the log and pattern F DF 110 times. Therefore $D \Rightarrow 2 W F = (89+110)/(89+110+1) = 0.995$. Because $F \notin C1$ and $D \notin C1$ and $0.995 \geq \sigma_{L2L}$ both $(F, D) \in C2$ and $(D, F) \in C2$. The same argumentation counts for the pattern EG resulting in $C2 = \{(F, D), (D, F), (E, G), (G, E)\}$.

4. Based on Table 4 check each non End-row for the highest value (the strongest follower). For example, for the C task the highest value (in boldface) is 0.997; therefore (C, I) is in the set Cout.

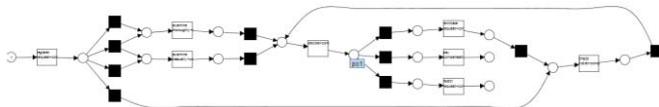
5. Based on Table 4 check each non Start-column for the highest value (the strongest cause). For example, for the K task the highest value (in boldface) is 0.998; therefore (H, K) is in the set Cin.

6, 7. As an illustration we take the tasks D and F. They are in a direct loop (i.e., $(D, F) \in C2$). The strongest output connection of D beside F is K (0.833), and from F is H (0.997). For this reason $(D, K) \in C0$ out (is not strictly necessary) and will be removed from Cout (step 7 of the algorithm). In Table 4 the removed connections are marked with underlining.

8,9. Analogue to step 6 and 7, but now for the incoming connections.

Table 4. All positive $a \Rightarrow_w b$ -values. See the example for a clarification of the bold face, Italic and underlined values.

	Start	A	B	C	D	E	F	G	H	I	J	K	L	End
Start	0	.999	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	.998 , .998	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	.031	.995 , .995	0	0	0	0	.323	.084	0	0	0
C	0	0	0	0	.328	.272	.421	.492	0	.997	0	0	0	0
D	0	0	0	0	0	0	<u>.650</u>	0	0	0	0	<u>.833</u>	.300	0
E	0	0	0	0	0	0	0	<u>.620</u>	0	0	0	0	.167	0
F	0	0	0	0	.993	0	0	.997	.0842	0	0	0	.667	0
G	0	0	0	0	.993	0	0	0	.997	.0232	0	0	.400	0
H	0	0	0	.15	0	0	0	0	0	.205	0	.998	0	0
I	0	0	0	0	.040	0	0	0	0	0	0	.998	0	0
J	0	0	0	0	.328	.395	.073	.054	.058	0	0	.997	.833	0
K	0	0	0	0	0	0	0	0	0	0	0	0	0	.999
L	0	0	0	.944	0	0	0	0	.267	0	0	0	0	0
End	0	0	0	0	0	0	0	0	0	0	0	0	0	0



[5] H. Mannila, H. Toivonen, and A.I. Verkamo. Discovery of Frequent Episodes in Event Sequences. *Data Mining and Knowledge Discovery*, 1(3), pages 259–289, 1997.

[6] J. Herbst and D. Karagiannis. Integrating Machine Learning and Workflow Management to Support Acquisition and Adaptation of Workflow Models. In *Proceedings of the Ninth International Workshop on Database and Expert Systems Applications*, IEEE, pages 745–752, 1998.

[7] Herbst and D. Karagiannis. An Inductive Approach to the Acquisition and Adaptation of Workflow Models. In M. Ibrahim and B. Drabble, editors, *Proceedings of the IJCAI'99 Workshop on Intelligent Workflow and Process Management: The New Frontier for AI in Business*, pages 52–57, Stockholm, Sweden, August 1999.

[8] L. Maruster, W.M.P. van der Aalst, A.J.M.M. Weijters, A. van den Bosch, and W. Daelemans. Automated Discovery of Workflow Models from Hospital Data. In B. Kröse, M. de Rijke, G. Schreiber, and M. van Someren, editors, *Proceedings of the 13th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2001)*, pages 183–190, 2001.

[9] Maruster, A.J.M.M. Weijters, W.M.P. van der Aalst, and A. van den Bosch. Process Mining: Discovering Direct Successors in Process Logs. In *Proceedings of the 5th International Conference on Discovery Science (Discovery Science 2002)*, volume 2534 of *Lecture Notes in Artificial Intelligence*, pages 364–373. Springer-Verlag, Berlin, 2002.

[10] *International Journal of Information Technology Convergence and Services (IJITCS) Vol.1, No.3, June 2011*

References

[1] Zeromskiego. U, “Fibres and Textiles in Eastern Europe”, October Vol 15, BO. 4(63), December 2007.

[2] W.M.P. van der Aalst and A.J.M.M. Weijters. *Process mining: a research agenda*. In *Computers in Industry 53*, Elsevier B.V., 2003,

[3] R. Agrawal, D. Gunopulos, and F. Leymann. Mining Process Models from Workflow Logs. In *Sixth International Conference on Extending Database Technology*, pages 469–483, 1998.

[4] H. Mannila and D. Rusakov. Decomposing Event Sequences into Independent Components. In V.Kumar and R. Grossman, editors, *Proceedings of the First SIAM Conference on Data Mining*, SIAM, pages 1–17, 2001.