ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN IMPROVED VIRUS EVOLUTIONARY GENETIC ALGORITHM FOR WORKFLOW MINING

CHENGFENG JIAN, FANG LI

College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, Zhejiang, China

ABSTRACT

Combined with virus evolutionary mechanism, the virus evolutionary genetic model algorithm which oriented workflow mining is put forward. It is based on the original main population that infection the population of individuals. First, in the workflow virus evolutionary mechanism introduction, it needs Petri net and causal relationship with matrix, the population of virus of individuals to chromosomes, setting the virus vitality and appealing. Second, the algorithm combines with the genetic algorithm in workflow mining on design. It designs a virus operators, infection operation and degradation operation. At the same time, the virus evolutionary genetic mode algorithm considers the virus itself has certain vitality. Experiments show that it can increase the diversity of the original population, and avoid premature convergence and precocious phenomena.

Keywords: *Workflow Mining, Causal Matrix, Hybrid Genetic Algorithm, Virus*

1. INTRODUCTION

How to optimize work flow and improve the efficiency of workflow becomes the concern focus of enterprises, so people put more and more attention on workflow mining. The aim of workflow mining is to establish the explicit workflow model. However current workflow mining algorithms mostly use local strategy, which cannot guarantee that a globally optimal workflow model is mined. In addition, these algorithms are not robust when logs contain noise. The genetic algorithms performing global search can overcome problems, and the problem of noise is naturally tackled by the genetic algorithms.

Aiming at the above questions, people pay more attention on the research of workflow mining based on genetic algorithm [1-6]. De Medeiros etc. proposed the genetic algorithm of workflow mining, this algorithm use global strategy to solve the problems, such as sequence, parallelism, selection, non-free selection structure and so on, and it is not sensitive about noise. SongHui etc. proposed the algorithm of workflow mining based on simulate anneal arithmetic. This algorithm can mine structures about a variety of workflow. But the use of genetic algorithm also exist defects in workflow mining, in the local search mainly displays premature convergence and precocious phenomena, often gets the non-optimal solution, cannot completely and accurately enough mine the model.

Based on the traditional genetic algorithm on the basis of workflow mining, this article puts forward a virus evolutionary genetic algorithm of workflow mining. Combined with virus evolutionary mechanism, it increases the diversity of population and the average fitted value, effectively avoids the loss of the optimal solution, speeds up the convergence and the speed of convergence, and effectively solves the problems about the genetic algorithms in the workflow mining application.

2. TRADITIONAL GENETIC ALGORIHTM

2.1 Basic Conceptions

Genetic Algorithm is a kind of learning method according simulation of biological evolution process, it attempt to simulate the actual problems to a population, according to certain rules code individuals of the population, the individual represents a solution, first initial population, and then according to the principle of survival of the fittest, evolved the optimal solution. Genetic algorithm is a new kind of calculation method, which combines and permeates nature genetics with computer science, so the genetic algorithm often uses some basic terms in the natural evolution.

Chromosome is known as the individual. As the most basic unit of genetic algorithms, it often represents form of a particular problem in the algorithm; generally, it is a set of binary number.

Gene is the smallest units of chromosomes, and in binary string represented chromosome, it usually shows through a digital. Code is a process that the solution of the problem is expressed as bit string. After coding, each bit string represents an individual, namely: a solution of the problem. Population is a certain number of individual constitute groups, namely: the set of solution; the number of individuals in the population is called population size. Fitness is an index about evaluating environmental adaptation ability of individual, namely, solution is good or bad. It comes from an evaluation function; the evaluation function is usually the objective function about solving problem, so it is also called adaptation function. Genetic operators can generate new individual, such as: selection, crossover and mutation, etc.

(1) Selection: Choose individual from population based on a certain probability. The operation is based on fitness; if the fitness of individual become higher, the probability about producing offspring becomes higher.

(2) Crossover: Exchange parts of gene in the two chromosomes, produce new individuals; Crossover probability decide the possibility of crossover operation on two individuals.

(3) Mutation: Randomly change part of gene in the chromosome.

2.2 Executing Procedure

The basic processes of genetic algorithm [6] show as follows:

(1) Coding: According to certain encoding rules, change the event log data into chromosome that can represent population individual.

(2) Initialization: Randomly choose N population individuals as the initial population $P(0)$ from the population, the initial iteration $t=0$, and the iterative number is T.

(3) Fitness: According to a fitness calculation function, calculate each individual's fitness value of population $p(t)$.

(4) Selecting operation: Based on the fitness, choose individual from population according to a certain probability.

(5) Interlace operation: According to certain cross probability, do crossover operations to the individual by choosing out.

(6) Mutation operation: According to certain probability, do mutation operation to the crossed population, and then produce the next generation $P(t+1)$ of population.

(7) Iterative termination judgment: if t is less or equal than T, $t=t+1$; if t is greater than T, the optimal fitness value getting from the iteration process is the optimal solution, then terminate the operation.

2.3 Characteristics of Genetic Algorithm

Genetic algorithm is a global search algorithm with robustness [7-11], features about it show as follows:

(1) Parallelism: Genetic algorithm can handle multiple individual among population at the same time, and evaluate multiple solutions in the search space, so the genetic algorithm has good global search capability, and reduce possible getting into local optimal solution.

(2) Only use fitness function to guide the search. Genetic algorithm only needs the fitness function about the coding series to guide the search, but other search methods generally need auxiliary information to work properly.

(3) Inner heuristic random searching capability. Genetic algorithm does not use deterministic rules, but use the change of the probability to guide the search direction.

(4) Genetic algorithm is easy to be putted in existing model, and has good scalability.

3. VIRUS EVOLUTIONARY GENETIC ALGORITHM

3.1 Encoding of Virus Chromosome

Because mining object is the event log, the event log is performed in order of activities; meanwhile, causal matrix can be better used to express the internal description form of the process that gets from the event log through mining, we using the causal matrix, activities out of logic and activities into the logic to describe the process model in this paper. Causal matrix show the relationship about predecessor and successor between activities, the matrix is $n\times n$. The n is the number of the activity, activities out of logic show the relationship about the current activity and the subsequent activity, activities into the logic show the logical relationship about the current activity and the subsequent activity. PM process model can be specific described as:

$$PM = (A_{set}, CM, O_{set}, I_{set}) \quad (1)$$

A_{set} is the set of activities, for example, $A_{set} = \{A_1, A_2, A_3 ... A_n\}$,the n stands for the number of activities, A_i stands for activities, CM stands for causal relationship, $CM = [C_{ij}]_{n \times n}$:

$$c_{ij} = \begin{cases} 1, A_i \text{ is associated with } A_j \\ 0, A_i \text{ is not associated with } A_j \end{cases} \quad (2)$$

If $c_{ij} = 1$, this means A_i is the precursor activity of A_j ,and that is called $A_i = \Pr eA(A_j)$,meanwhile, A_j is the subsequent activity of A_i ,and that is called $A_j = \mathrm{P}\, ostA(A_i)$. If $c_{ij} = 0$, this means there has been no contact between the two activities.

O_{set} stands for output logic of activity, for example, $O_{set} = \{O_1, O_2, O_3 ... O_n\}$,the n stands for the number of activities, O_i stands for output logic of A_i , O_i can be describe as { A_j / A_k }, j is not equal to k, A_j and A_k are subsequent activities of A_i .

I_{set} stands for input logic of activity, for example, $I_{set} = \{I_1, I_2, I_3 ... I_n\}$,the n stands for the number of activities, I_i stands for input logic of A_i , I_i can be describe as{ A_j / A_k }, j is not equal to k, A_j and A_k are precursor activities of A_i .

3.2 Fitness

3.2.1 Fitness of individual

Combined with integrity and accuracy of genetic individual, the following give the definition of genetic individual fitness:

Suppose R is event log of workflow, CM is causal matrix, CM [] is a muster about the set of causal matrixes and the definition of function about individual fitness.

$$fitnessHost = PF_{complete}(R, CM) - k * PF_{precise}(R, CM, CM[])$$

Fitness function fitnessHost measure redundant activities of genetic individuals through the right value k, we can see from the definition, that if the integrity of workflow logs which genetic individual analyses out is higher, and the redundant activities outside of logs is less, the value of fitness function is higher.

3.2.2 Fitness of virus

According to the workflow mining and the main characteristics of population individuals, the fitness function of virus population individual that the paper designs is: A virus population individual can infect 1 or more main population individuals, and the main population individuals that are infected must be randomly selected in accordance with the Infection probability, when the main population individuals are infected, the value of its fitness function will change at the same time; the fitness function of a virus population individual just is the value of fitness function about the main population isn’t infected minus the value of fitness function about the main population had been infected; According to the design of the main population individual in this paper, the more higher the value of fitness, the more better the main population individual, so, if the more bigger the value of fitness function about Virus population individuals, the more bigger the promoting effect about this virus population individuals to the main population individual role.

For a virus population individuals i, muster T is a muster about the main population set of individuals that are infected by the virus population individuals, for any individual k during the main population T, using $fitnessHost_k$ stands for the value of fitness function before the main population individual k is infected, using $fitnessHost_k{}'$ stands for the value of fitness function after the main population individual k is infected, then the fitness function of this virus individual I is described such as formula 2-2 follows:

$$fitnessVirus_i = \sum_{k \in T} (fitnessHost_k' - fitnessHost_k) \quad (3)$$

3.3 Vitality

The reason of virus can survive is it can play a role in promoting on the evolution of population individuals, so, the existence of virus population individual has a life cycle, the life cycle can be show with vitality; if a virus population individuals has died or lost vitality, the individual should be removed and be replaced by the new virus population individual.

The t+1 generation of virus population individuals i, its vitality am expressed as:

$$life_{i,t+1} = r \times life_{i,t} + fitnessVirus_i \quad (4)$$

3.4 Infectivity

When the virus population individuals infect the main population individuals, according to certain probability we random select a virus population individual i to infect the main population; for the inflection ability of virus population individuals i, we use probability $P_{\inf ect}$ to express it in this paper, the specific definition is listed as follows:

$$P_{\inf ect} = \min\{(1+a_{i,t})P_{initInfect}, P_{\max Infect}\} \quad (5)$$

3.5 Operation of virus

According to the infectivity $P_{\inf ect}$,the virus population individuals infect the main population individuals, and create new offspring, use the genes of virus population individual to replace the corresponding genes of main population individual, the specific operation are as follows:

First, we choose the main population individuals that would be infected, then according to the probability of infection $P_{\inf ect}$, decide whether do infection operation to the individuals. Next we choose a crossing for carrying out the infection of virus individual and the main population individuals, and exchange some of the structure of the main population individual before or after the point, then create a new individual. In this paper each the main population individual or virus individual corresponds to a workflow model, the model is expressed through causal matrix, then according to the probability of infection $P_{\inf ect}$,the two causal matrixes $C = (T, CM, I, O)$ do operation of infection, a couple of T are determined and uniform, so T is not to participate in the infection. According to the constraints of CM, I and O, CM can be deduced from I and O, so the infection method is: we random select a task t as infection point, separately do single point infection to input set I and output set O of the task, namely, random select a subset as a substitution point between input set I and output set O at the same time, then replace from start subset of infection to the end subset of infection; finally, we examine input set I and output set O of the new main population, make sure that the new main population get the correct infection, and avoid coming inconsistency about causal matrixes. The pseudo codes about the operation of infection are as follows:

Input: The infection rate of main population host and virus $P_{\inf ect}$.

Output: newHost

Firstly, give n assignment, then, make sure newHost is equal to host, and newVirus is equal to virus.

Secondly, according to the current infection rate, carry out:

(1) Random select a task t as substitution point.
(2) Separately random select a substitution point sp1and sp2 for the input sets InnewHost(t) and InnewVirus(t) of task t in newHost and newVirus, substitution $po\text{int} \in [0, n-1]$,the n stands for the number of subset about input conditions function I.
(3) Establish set: remainSet1 stands for a subset from the first subset to replace point set, is not included in the replacement point set, swapSet1 stands for a subset from the switch point to the end of polymerization, remainSet2 and swapSet2 express the meaning of equal.
(4) Random generate the random number r that its value is the number from 1 to 9, according to the value of r, for each subset of swapSet2, we do the following operation.
    a. If r is equal or greater than 1 and r is less than or equal to 3, S2 will be added to remainSet1 as a new subset.
    b. If r is greater than 3 and r is less than or equal to 6, activities in S2 will be added to a subset in remainSet1.
    c. If r is greater than 6 and r is less than or equal to 9, a subset R1 in remainSet1 is selected, the activities those exist in R1 and at the same time t also exist in S2, then S2 is added to remainSet1.

Thirdly, for swapSet1 and remainSet2, execute Step4.

(1) Give anew InnewHost assignment, then, make sure InnewHost is equal to remainSet1, give anew InnewVirus assignment, then, InnewVirus is equal to remainSet2.
(2) Output function O take the place of input function I, then execute Steps 2 from to 6.
(3) Update tasks are closely related to t.

Fourthly, return newHost. After operations of infection are completed, a task may not appear in another task corresponding I/O, but the corresponding position of the corresponding relationship matrix about CM is 1, so, after the operation is completed, checking the I/O get from exchange is reasonable or not, whether there is a problem causal matrixes aren’t consistent. If there is inconsistency, CM and causal matrix should be adjusted and updated.

In addition, there is another operation called copy, namely, virus population individuals copy gene substring that has certain length from the main population individuals to generate a new virus, this operation is similar to the operation of infection, the difference only is using the main population to infect the virus population; another operation is the operation of involution, the initial virus population individuals become vestigial through random selecting active point in the individual.

Involution is refers to change existing causal relationship in the virus group, namely, according to the degradation probability P_{cut} , we can perform the following actions in individuals: random selecting a subset from input/output function of an active in the virus individuals, and adding an active which come from the set of activities to the subset; random selecting a subset, and deleting an activity from the subset; the elements in input/output function of the subset random are distributed to the new subset again; The pseudo codes about the operation of involution are as follows:

Input: virus individual, degradation probability
Output: the degenerate virus individual

(1) For the degenerate probability and virus individual, performing the following operation: For a task t of the individual virus, we take out the corresponding I and O. Random select a subset S from I, then generate the random number r that its value is the number from 1 to 9.If r is greater than 0 and r is less than or equal to 3, the task will be remover. If r is greater than 3 and r is less than or equal to 6, random selecting a task t1 from muster T, and then t1 is inserted into a subset S. If r is greater than 6 and r is less than or equal to 9, the elements in input/output function of the subset random are distributed to the new subset again.
(2) O is replaced by I, then repeating B in step 1.
(3) Update tasks are closely related to t.
(4) Return the degenerate virus individual

3.6 Steps of the Algorithm

The steps of virus evolutionary genetic algorithm are as follows:

Step 1, randomly generate a certain amount of scale for N of the initial value, which constitute the main group, and then produce a virus initial group by a certain probability.

Step 2: do interlace operation on the main group of individuals by a certain probability (PC), then use the new individual to replace the father generation individual, individuals of the main group will change by a probability (PM).

Step 3: respectively take out of the individual in virus group in corresponding to the individual in main group, and then infect the main group by a certain probability.

Step 4: calculate the adaptive value of each main groups, and calculate the infection and vitality of virus.

Step 5: according to the adaptive value, choose good individual to consist a new generation of the same scale, and then choose new individual based on appeal and vitality.

Step 6: repeat step 2 ~ 5 steps, until meet the condition of end.

4. SIMULATION AND ANALYSIS OF THE RESULTS

4.1 Simulation of the System

Based on the above algorithm, we have developed a corresponding workflow mining platform. Two algorithms are designed in the paper: the first algorithm is the traditional genetic algorithm-based workflow mining; the second algorithm is the new combination of virus evolution mechanism of the genetic algorithm-based workflow mining which is described in this paper. They are separately called GA and VGA for short.

Experimental data, event logs are collected from the log set of process mining website. This log set contains 23 event logs, in which each event log represents a different kind of workflow model. So, there are 23 kinds of the original models. These models contain such structures: sequence, selection, non-free choice, parallel, loops, implied tasks and so on. The model numbers of Logs about these events representative have a large number of tasks and are relatively complex, very close to the real model. This study developed by NET technology and SQL SERVER2005 database.

4.2 Analyses of Experimental Results

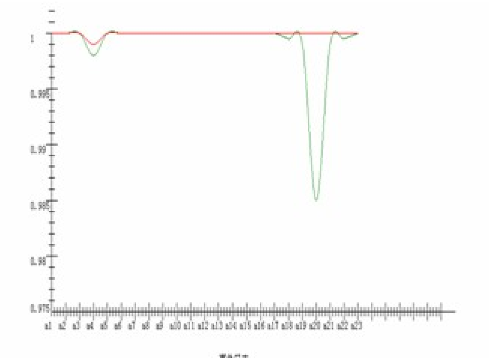

Figure 1: 23 A Log Of GA And VGA Comparison

Through the experimental results in Figure 1, we know that 23 log models exhumed by the VGA algorithm have higher integrity than the 23 log mining models using the GA algorithm. The integrity of the model using VGA algorithm tends to 1 in addition to the g4, which is higher than the GA mining model obtained.

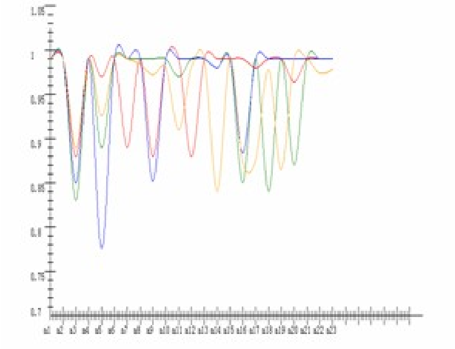

Figure 2: GA Algorithm To 23 Log Mining Model Behavior/Structure Precision And Repeatability Degrees

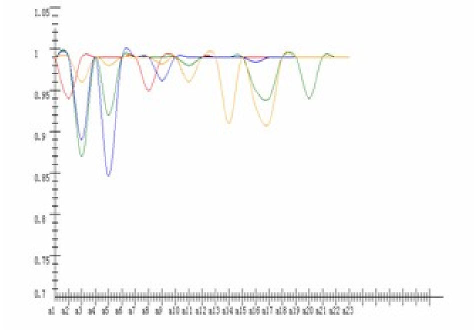

Figure 3: VGA Algorithm To 23 Log Mining Model Behavior/Structure Precision And Repeatability Degrees

By comparison of Figures 2 and 3, we clearly see that the behavior or structure about accuracy of the model and reproducibility degrees of 23 logs mined by VGA algorithm is higher than the behavior of th e GA algorithm did.

5. CONCLUSION

This article gives some improvements on the local search of genetic algorithm-based workflow mining and uses this algorithm in digging out the workflow model, which improves accuracy and integrity. However, there are some shortcomings. First, workflow mining algorithms that this paper designed and researched cannot take factors into account in some event log, such as the process increments problems in the event log and time factor, which are some factors that practical projects should consider. Second, Local search algorithm can also be combined with other evolutionary mechanisms so that we can improve on the genetic algorithm-based workflow mining.

REFERENCES:

[1] V. Vander, B. Dorgan, "Workflow mining: discovering process models from event logs", IEEE Transactions on Knowledge and Data Engineering, August 20-23, 2004, pp. 1128-1142.

[2] T. Murata, "Petri Nets: Properties, analysis and applications", Proceeding of The IEEE, May 13-17, 2006, pp. 541-580.

[3] G. Schimm, "Process miner-a tool for mining process schemes from event-based data", *Lecture Notes in Computer Science*, Vol. 24, No. 24, 2002, pp. 525-528.

[4] J. Herbst, D. Karagiannis, "Workflow mining with inWoLvE", *Computers in Industry*, Vol. 13, No. 53, 2004, pp. 151-162.

[5] A. Rozinat, W. Pvan, "Dicision mining in ProM", *Lecture Note in Computer Science*, Vol. 7, No. 102, 2006, pp. 420-425.

[6] A. Lim, C. Lee, M. Raman, "Hybrid genetic algorithm and association rules for mining workflow best practices", Expert Systems with Applications, Vol. 39, No. 9, 2012, pp. 10544-10551.

[7] P. Zhang, N. Serban, "Visualization and performance analysis of enterprise workflow", *Computational Statistics and Data Analysis*, Vol. 3, No. 51, 2007, pp. 267-269.

[8] A. Medeiros, M. Weijters, "Working Mining: Current status and future direction", *Lecture Notes in Computer Science*, Vol. 9, No. 7, 2010, pp. 389-401.

[9] K. Alves, B. Fvan "An overview and a concrete algorithm", *Lecture Notes in Computer Science*, Vol. 2, No. 7, 2004, pp. 154-168.

[10] W. Pvan, M. Weijters, "Process raining: a research agenda", *Computers in Industry*, Vol. 53, No. 3, 2010, pp. 241-244.

[11] L. Wen, J. Wang, "Implicit dependencies between tasks from event logs", *Lecture Note in Computer Science*, Vol. 30, No. 35, 2006, pp. 591-603.