Modeling and Performance Analysis of Workflow Based on Advanced Fuzzy Timing Petri Nets

Huifang Li and Xinfang Cui

Abstract Time management plays an important role in workflow management systems. Aiming at describing all the uncertain time information existing in practical business process, this paper applied fuzzy sets theory to workflow time modeling and performance analysis and then presented an improved fuzzy timing workflow nets. Based on the reduction analysis of fuzzy time performance for several constructions of workflow models, a fuzzy time analysis and reasoning method was proposed and then a hierarchical algorithm for business process temporal reachable probability analysis was proposed. Finally, a case study illustrates that our method is feasible and efficient.

Keywords Fuzzy time constraint · Workflow · Possibility theory · Advanced fuzzy timing workflow nets

1 Introduction

Workflow modeling and model performance analysis are important research contents of workflow. In actual business processes, there are many time constrains, and if violated, it may bring loss to the enterprise, so the timing description ability of workflow models becomes the focus of current workflow modeling studies [1].

Aalst first applied Petri net, which has intuitive graphical representation and solid mathematical foundation, to workflow management and proposed workflow net (WF-net) [2]. Recently, in order to describe and analyze time behaviors in

H. Li (✉) · X. Cui
Automation School, Beijing Institute of Technology University, Beijing, China
e-mail: huifang@bit.edu.cn

X. Cui
e-mail: cuixinfangxiaoxin@163.com

F. Sun et al. (eds.), *Foundations and Applications of Intelligent Systems*,
Advances in Intelligent Systems and Computing 213,
DOI: 10.1007/978-3-642-37829-4_17,

workflow, many Petri net-based timing workflow models are proposed, such as time workflow net(TWF-net) by Ling and Schmidt [3], and timing constraint workflow nets (TCWF-net) by Li and Fan [4]. In these models, time information is certain; however, resources and activities have dynamic characteristics in actual workflow because of the uncertain factors, such as machine fault, different proficiency of workers and so on, so there is a lot of uncertain time information which is hard to describe and analyze. Aiming at this requirement, Murata first applied fuzzy theory to time Petri nets and proposed Fuzzy-timing High-level Petri nets (FTHN) [5]. Pan and Tang [6] added certain effective time constraints on resources and activities and proposed fuzzy temporal workflow nets (FTWF-nets). But in order to improve the flexibility and stability of the dynamic workflow management of real business processes, the effective time constraints attached to resources and activities should have fuzzy values as well. So based on FTWF-nets, by applying high-level fuzzy timed Petri net (HFTN) [7] to workflow modeling, the advanced fuzzy timing workflow nets (AFTWF-nets) model was proposed in this paper and a stratification algorithm of temporal reachable probability of AFTWF-nets model was given.

In AFTWF-nets model, fuzzy time was used to describe the effective time constraints of resources and activities, calculate, and analyze. Meanwhile, the stratification algorithm of temporal reachable probability based on AFTWF-nets reduced the complexity of the data structure and made it much easier to develop and realize the workflow management system.

2 Advanced Fuzzy Timing Workflow Nets

2.1 Fuzzy Time Concept

Definition 1 Fuzzy time point is the possibility distribution of a function mapping from the time scale Γ to real interval [0, 1], which restricts the possible value of a time point. Let π_a denotes the possibility function attached to a time point a, then $\forall\tau \in \Gamma, \pi_a(\tau)$ denotes the numerical estimate of the possibility that a is precisely τ. Let fuzzy set A be the possible range of a, and μ_A denote the membership function of A and then we have $\forall\tau \in \Gamma, \pi_a(\tau) = \mu_A(\tau)$. In this paper, a fuzzy time point is denoted by trapezoid possible distribution [6], which must be normal and convex. So a fuzzy time point can be represented by $h[\pi_1, \pi_2, \pi_3, \pi_4], (\pi_1 \leq \pi_2 \leq \pi_3 \leq \pi_4)$ (Fig. 1 shows an example). And it becomes a fixed-length time when $\pi_1 = \pi_2, \pi_3 = \pi_4$ and a fixed time point when $\pi_1 = \pi_2 = \pi_3 = \pi_4$.

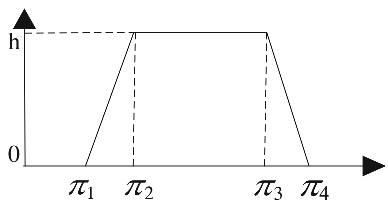


Fig. 1 Trapezoid function of fuzzy time point

2.2 AFTWF-nets Model

Definition 2 AFTWF-nets is a 8–tuple $(P,\ T,\ F,\ M_0,\ \mathrm{FT},\ \mathrm{FD},\ \mathrm{FTPC},\ \mathrm{FTTC})$.

1. $(P,\ T,\ F,\ M_0)$ is a basic workflow net. P is a set of places, T is a set of transitions, $P \cap T = \emptyset, P \cup T \neq \emptyset$. F is a set of arcs which is a subset of $(P \times T) \cup (T \times P)$. M_0 represents the initial marking. Let t denotes a transition, and p denotes a place.
2. FT, fuzzy timestamp, denotes a set of fuzzy timestamps attached to tokens, which represents the possibility distribution of a token's arrival time at a place, and let $\pi(\tau) = h[\pi_1, \pi_2, \pi_3, \pi_4]$ denotes the fuzzy timestamps function.
3. FD, fuzzy delay, denotes a set of fuzzy delays of transitions, which describes the possibility distribution of the duration from t trigging to t outputting tokens to its output places, and let $d_t(\tau) = d[d_1, d_2, d_3, d_4]$denotes the fuzzy delay function.
4. FTPC, fuzzy token time constraint, denotes a set of valid intervals constraints of tokens, which is signed as FTPC $(p) = 1[a_1, a_2, a_3, a_4]$. Let$\tau$ denotes a time point when a token arrives at a place, so the token is definitely valid during time interval $[\tau + a_2, \tau + a_3]$, it is uncertainly valid in $[\tau + a_1, \tau + a_2]$, and it is definitely invalid out of $[\tau + a_1, \tau + a_4]$. If FTPC (p) is a fixed time interval, then $a_1 = a_2$ and $a_3 = a_4$. If there is no time constraint, then FTPC $(p) = 1[0,\ 0,\ 0,\ 0]$.
5. FTTC, fuzzy transition time constraint, denotes a set of valid intervals constraints of transitions, which is signed as FTTC $(t) = 1[b_1,\ b_2,\ b_3,\ b_4]$. Let τ denotes a time point when a transition is enabled, so the transition surely can trigger in $[\tau + b_2, \tau + b_3]$, it possibly triggers in $[\tau + b_3, \tau + b_4]$, and it surely cannot trigger out of $[\tau + b_1, \tau + b_4]$. If FTTC (t) is a fixed time interval, then $b_1 = b_2$ and $b_3 = b_4$. If there is no time constraint, then FTTC $(t) = 1[0,\ 0,\ 0,\ 0]$.

In this model, FD denotes the fuzzy delays of transitions, FTPC limits the life cycle of resources, and FTTC limits the fire time interval of transitions. Such model lays the modeling foundation for the following expanded logical analysis as well as the time-level performance optimization and unification.

3 Performance Analysis of AFTWF-nets

3.1 Time Calculation of AFTWF-nets

Murata. T has given the algorithm of FTN [5], and based on this, AFTWF-nets model redefined and remodeled the time constraint of tokens and transitions, so the new definitions and detailed calculations of time algorithm are shown as follows. In AFTWF-nets, $I_p(t)$ represents input places set of t, $O_p(t)$ represents output places set of t. p_i is the initial place, and p_o is the ending place. In order to facilitate the description and discussion, the workflow net is assumed to be reasonable.

Definition 3 Fuzzy enabled time $e_t(\tau)$: fuzzy enabled time of t denotes the possibility distribution of that t is enabled at time point τ, and $e_t(\tau)$is decided by the possibility distribution of the latest arrival time of the tokens in $I_p(t)$.

If there is only one token in $I_p(t)$, and it arrives at $\pi(\tau)$, so $e_t(\tau) = \pi(\tau) \oplus \mathrm{FTPC}(p)$

$$\begin{aligned} &= \min\{h,\ 1\}[\pi_1,\ \pi_2,\ \pi_3,\ \pi_4] \oplus [a_1,\ a_2,\ a_3,\ a_4] \\ &= h[\pi_1 + a_1, \pi_2 + a_2, \pi_3 + a_3,\ \pi_4 + a_4]. \end{aligned}$$

where: $\oplus$ is extended additive operator [6].

If there are n tokens in $I_p(t)(p_1,\ p_2 \ldots p_{\mathrm{n}})$, token in p_i arrives at $\pi_i(\tau) = h_i[\pi_{i1}, \pi_{i2}, \pi_{i3}, \pi_{i4}]$ and $\mathrm{FTPC}(p_i) = 1[a_{i1},\ a_{i2},\ a_{i3},\ a_{i4}]$. Then, $e_t(\tau) =$latest$\{\pi_i(\tau) \oplus 1[a_{i1},\ a_{i2},\ a_{i3},\ a_{i4}]\} = \min\limits_i\{h_i\}[\max\limits_i\{\pi_{i1} +\ a_{i1}\}, \max\limits_i\{\pi_{i2} + a_{i2}\}, \max\limits_i\{\pi_{i3} + a_{i3}\}, \max\limits_i\{\pi_{i4} + a_{i4}\}], i = 1,\ 2, \cdots n.$

where: latest operator is the trapezoidal function approximation algorithm.

Definition 4 Fuzzy occurrence time $o_t(\tau)$: fuzzy occurrence time of transition t denotes the possibility distribution of that t is fired at time point τ.

If there is no structural conflict, and FTTC $(t) = 1[b_1,\ b_2,\ b_3,\ b_4]$, then $o_t(\tau) = e_t(\tau) \oplus 1[b_1,\ b_2,\ b_3,\ b_4]$, else there is structural conflicts between m enabled transitions $(t_1,\ t_2 \ldots t_{\mathrm{m}})$, t_i is enabled at $e_{\mathrm{ti}}(\tau) = e_i[e_{i1},\ e_{i2},\ e_{i3},\ e_{i4}]$, $\mathrm{FTTC}(t_i) = 1[b_{i1},\ b_{i2},\ b_{i3},\ b_{i4}]$. In order to facilitate discussion, the situation where there is structural conflict between enabled transitions follows "first come first serve" strategy, which means the earlier enabled transition has higher priority. So for $t_j,\ e_{t\mathrm{j}}(\tau) = e_j[e_{j1}, e_{j2}, e_{j3}, e_{j4}]$, then

$$\begin{aligned} o_{\mathrm{tj}}(\tau) &= \mathrm{MIN}\{e_{\mathrm{tj}}(\tau) \oplus 1[b_{j1},\ b_{j2},\ b_{j3},\ b_{j4}],\ \mathrm{earliest}\{e_{\mathrm{ti}}(\tau) \oplus 1[b_{i1},\ b_{i2},\ b_{i3},\ b_{i4}]\}\} \\ &= \mathrm{MIN}\{e_j[e_{j1},\ e_{j2},\ e_{j3},\ e_{j4}] \oplus 1[b_{j1},\ b_{j2},\ b_{j3},\ b_{j4}], \\ &\quad\ \mathrm{earliest}\{e_i[e_{i1},\ e_{i2},\ e_{i3},\ e_{i4}] \oplus 1[b_{i1},\ b_{i2},\ b_{i3},\ b_{i4}]\}\} \\ &= \mathrm{MIN}\{e_j[e_{j1} + b_{j1},\ e_{j2} + b_{j2}, e_{j3} + b_{j3},\ e_{j4} + b_{j4}], \\ &\quad \max_i\{e_i\}[\min_i\{e_{i1} + b_{i1}\}, \min_i\{e_{i2} + b_{i2}\}, \\ &\quad \min_i\{e_{i3} + b_{i3}\},\ \min_i\{e_{i4} + b_{i4}\}],\},\ i = 1, 2, \cdots m. \end{aligned}$$

where: the earliest operator means to pick up the earliest enable time from m enable time points and is the trapezoidal function approximation algorithm. MIN is the operator to seek the intersection of possibility distributions.

Definition 5 Fuzzy execution delay $d_t(\tau)$: fuzzy execution delay of transition t is attached on the output arc of t, which denotes the possibility distribution of that it costs τ to finish t. If $o_t(\tau) = o[o_1, o_2, o_3, o_4]$, and $d_t(\tau) = d[d_1, d_2, d_3, d_4]$, then token arrive at $O_p(t)$ at $\pi(\tau) = o_t(\tau) \oplus d_t(\tau) = \min\{o,\ d\}[o_1 + d_1,\ o_2 + d_2,\ o_3 + d_3,\ o_4 + d_4]$.

3.2 Time Performance Analysis of AFTWF-nets

In AFTWF-nets, possibility theory is applied to describe time information and analyze time performance. Possibility operator and performance index are defined as follows.

Definition 6 Possibility function [6]: Suppose a and b are fuzzy time points, their possibility distributions are represented by trapezoidal function $\pi_a(\tau) = 1[a_1, a_2, a_3, a_4]$, which is signed as ABCD, and $\pi_b(\tau) = 1[b_1, b_2, b_3, b_4]$, which is signed as EFGH, as shown in Fig. 2. Then, the possibility that fuzzy time b is before a (signed as $b \le a$) can be: Possibility $(b \le a) = \frac{\text{Area}([a,b] \cap b)}{\text{Area}(b)} = \frac{\text{Area(EBCH)}}{\text{Area(EFGH)}}$. Especially, if b is a certain time, as shown in Fig. 3, $\pi_b(\tau) = 1[t, t, t, t]$, then $\text{Possibility}(a \le b) = \frac{\text{Area(AEFD)}}{\text{Area(ABCD)}}$.

Definition 7 Activities execution time constraint satisfaction is the possibility that an activity's execution time meets the expected time, which means the activity ends within the expected time. In AFTWF-nets, assume a token arrives at p_{i} at $\pi_i(\tau)$ and expect the transition t finishes at $\pi(\tau)$. The token begins from p_{i}, goes through a series of transitions and finally arrives at $O_p(t)$ at $\pi_a(\tau)$, which can be worked out using the timing algorithm in Sect. 3.1. So the activities execution time constraint satisfaction is Possibility $(\pi_a(\tau) \le \pi(\tau))$.

Definition 8 Time distance between activities constraints satisfaction is the possibility that the time distance between two activities is no less than or no more than the expected time distance. In this paper, it is represented by the difference of

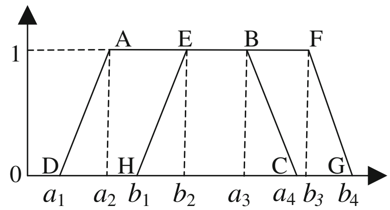


Fig. 2 $a \le b$(b is a fuzzy time point)

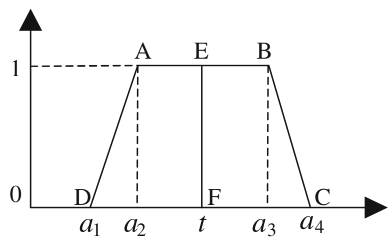


Fig. 3 $a \le b$(b is a certain time point)

activities' fuzzy occurrence time. Assume transition t_1 fires before transition t_2, and the expected time distance is $D(\tau)$. After calculation, we get $o_A(\tau)$ and $o_B(\tau)$, then the possibility that the time distance between t_1 and t_2 is no less than $D(\tau)$ is Possibility $(\pi_a(\tau) \le \pi(\tau))$ and the possibility that the time distance between t_1 and t_2 is no more than $D(\tau)$ is Possibility$(o_B(\tau) \le o_A(\tau) \oplus D(\tau))$.

Definition 9 Temporal reachable probability of process is the possibility that a workflow process finishes within expected time distance. Assume a token arrives at p_i at $\pi_i(\tau)$, and the expected time distance is $D(\tau)$, which means the token is expected to arrive at p_o at $\pi_i(\tau) \oplus D(\tau)$. The token begins from p_i, goes through a series of transitions and finally arrives at p_o, then we can work out $\pi_o(\tau)$. So the temporal reachable probability of process is Possibility$(\pi_{\mathrm{po}}(\tau) \le \pi_{\mathrm{pi}}(\tau) \oplus D(\tau))$.

3.3 A Stratification Algorithm of Temporal Reachable Probability

In this paper, a stratification algorithm of temporal reachable probability analysis in AFTWF-nets is proposed, which stratifies and extracts AFTWF-nets into hierarchical advanced fuzzy timing workflow nets (HAFTWF-nets) based on the four control route structures of workflow, which are sequence, selecting, parallel, and circle route. Here are the rules of transforming AFTWF-nets to HAFTWF-nets as follow.

1. For sequence route, merge the sequence route without any branch or loop node, represent it in the form of a group of subnets, and represent it by a "place->transition->place" structure named "sequence" in the original net.
2. For parallel route, represent all concurrent branches between "And split" and "And join" in the form of a group of subnets and represent it by a "place->transition->place" structure named "parallel" in the original net.
3. For selecting route, represent all conditional branches between "Or split" and "Or join" in the form of a group of subnets and represent it by a "place->transition->place" structure named "select" in the original net.
4. For circle route, represent the loop in the form of a group of subnets, and represent it by a "place->transition->place" structure named "circle" in the original net.
5. Each route structure can be nested within each other.

Based on the above transforming rules, the concrete steps of stratification algorithm are shown as follows (Fig. 4):

Step 1: Visit p_i.

Step 2: Visit the following nodes successively. When there is parallel, selecting or circle structure, extract the sequence structure before them, then visit their join nodes, do step 3; when visiting p_o, do step 4.

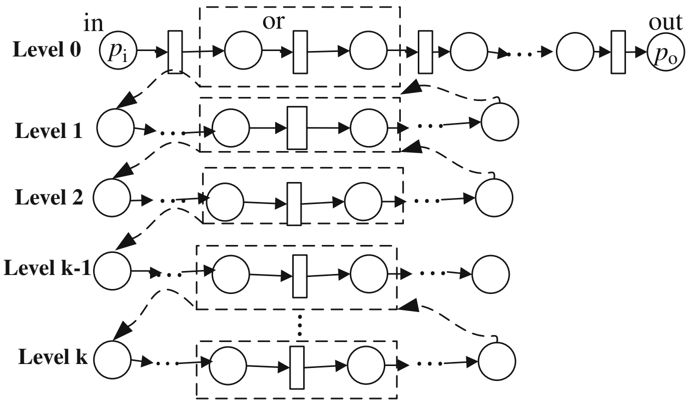


Fig. 4 AFTWF-nets to HAFTWF-nets

Step 3: Extract parallel, selecting and circle structure in subnet form and return to step 2.

Step 4: End visiting. Use the above algorithm in dealing with every subnet successively until there is only sequence structure after simplification.

AFTWF-nets model is called original net, and after transforming, it is called level 0 subnet. Assume the bottom is level k subnet, which contains four basic routes, so when calculating temporal reachable probability, it is needed to calculate level 0 subnet. The simplification and calculation of the basic route structures are shown as follows:

Sequence route: Assume there are n transitions t_1, t_2,…t_n in sequence pattern. So when representing this subnet shown in Fig. 5a as b with p_s, t_s, p_{s+1}, the token in p_{s+1} arrives at: $\pi_{ps+1}(\tau) = \pi_{ps}(\tau) \oplus \sum_{i=1}^{n}{}^{\oplus}(\mathrm{FTPC}(p_i) \oplus \mathrm{FTTC}(t_i) \oplus d_{ti}(\tau))$.

where $\sum_{i=1}^{n}{}^{\oplus}$denotes continuous extended plus, $\pi_{ps}(\tau) = \pi_{p1}(\tau), i = 1, 2, \cdots n$.

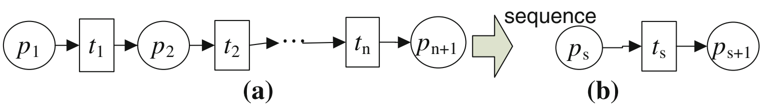


Fig. 5 Simplification of sequence route

Parallel route: Assume there are n transitions t_1, t_2,…t_n in parallel pattern. t_{as} is "And split", t_{aj} is "And join", $p_{i1} \in I_p(t_i,)$ and $p_{i2} \in O_p(t_i,)$. So when representing this subnet shown in Fig. 6a as b with p_p, t_p, p_{p+1}, the token in p_{p+1} arrives at:

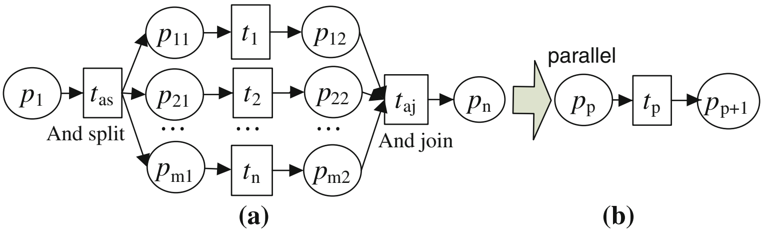


Fig. 6 Simplification of parallel route

$$\pi_{pp+1}(\tau) = \pi_{pp}(\tau) \oplus \text{FTPC}(p_p) \oplus \text{FTTC}(t_{as}) \oplus d_{tas}(\tau) \oplus \text{latest}\{\sum_{i=1}^{n}{}^{\oplus}(\text{FTPC}(p_{i1}) \oplus \text{FTTC}(t_i) \oplus d_{ti}(\tau) \oplus \text{FTPC}(p_{i2}))\} \oplus \text{FTTC}(t_{aj}) \oplus d_{taj}(\tau).$$

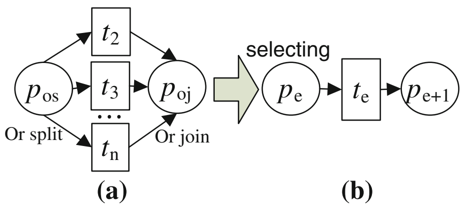
Where $\pi_{pp}(\tau) = \pi_{p1}(\tau),\ \text{FTPC}(p_p) = \text{FTPC}(p_1), i = 1, 2, \cdots n.$

Selecting route: Assume there are n transitions t_1, t_2,…t_n in selecting pattern. p_{os} is "Or split", p_{oj} is "Or join". And the possibility of selecting branch i, which contains t_i, is P_i, where $\sum P_i = 1$. So when representing this subnet shown in Fig. 7a as b with

$$p_e, t_e, p_{e+1},\ \text{the token in } p_{e+1} \text{arrives at} : \pi_{pe+1}(\tau) = \sum_{i=1}^{n} (P_i \times (\text{FTPC}(p_e) \oplus \text{FTTC}(t_i) \oplus d_{ti}(\tau)))$$

where $\pi_{pe}(\tau) = \pi_{pos}(\tau), FTPC(p_e) = FTPC(p_{os}), i = 1, 2, \cdots n.$

Fig. 7 Simplification of selecting route



Circle route: Assume there are two transitions t_1, t_2, in circle pattern it is P_1 to continue executing other transitions and it is P_2 to return to execute t_2, and $P_1 + P_2 = 1$. The loop which contains t_2 has been executed s times, so when representing this subnet shown in Fig. 8a as b with subnet as p_c, t_c, p_{c+1}, the token in p_{c+1} arrives at:

$$\pi_{pc+1}(\tau) = (1 + \sum_{i=1}^{s} p_2^s) \times (\text{FTPC}(p_c) \oplus \text{FTTC}(t_1) \oplus d_{t1}(\tau)) \oplus \sum_{i=1}^{s} p_2^s \times (\text{FTPC}(p_2) \oplus \text{FTTC}(t_2) \oplus d_{t2}(\tau))$$

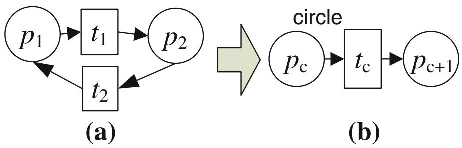


Fig. 8 Simplification of *circle* route

After Simplification, we can calculate the fuzzy time of each route in level k and then drag it to upper level $k-1$ to calculate. After k times iteration until working out the fuzzy time level 0, it is temporal reachable probability of the whole workflow process. Through the stratification, calculation process is simplified, as well as the probability of state explosion is reduced.

4 Case Study and Analysis

A company receives an order and arranges for the procurement and production. Now, the concrete steps are shown as follows. (Fig. 9 shows its AFTWF-nets model of the business process, and Table 1 shows the significances and time information of transitions and places):

The order arrives at p_i at $\pi_{\mathrm{pi}}(\tau) = 1[0,\ 0,\ 0,\ 0]$, time calculation is shown as follows:

Step 1: t_1. As we known$e_{t1}(\tau) = 1[0,\ 0,\ 0,\ 0]$, $\mathrm{FTTC}(t_1) = 1[0,\ 0,\ 0,\ 0]$ so $o_{t1}(\tau) = 1[0,\ 0,\ 0,\ 0]$.

Step 2: p_1, p_2 and p_{12}. p_1, p_2, $p_{12} \in I_p(t_1)$ so $\pi_{p1}(\tau) = \pi_{p2}(\tau) = \pi_{p12}(\tau) = o_{t1}(\tau) \oplus d_{t1}(\tau) = 1[2, 3, 5, 6]$.

Step 3: t_2, p_3, t_4, p_5, t_6 and p_7. $\mathrm{FTPC}(p_1) = 1[0,\ 0,\ 0,\ 0]$, $\mathrm{FTTC}(t_2) = 1[0,\ 0,\ 0,\ 0]$, $e_{t2}(\tau) = 1[2,\ 3,\ 5,\ 6]$, so $o_{t2}(\tau) = 1[2,\ 3,\ 5,\ 6]$, $\pi_{p3}(\tau) = o_{t2}(\tau) \oplus d_{t2}(\tau) = 1[3,\ 5,\ 8,\ 10]$, $e_{t4}(\tau) = \pi_{p3}(\tau) \oplus \mathrm{FTPC}(p_3) = 1[3,\ 5,\ 8,\ 10]$, $o_{t4}(\tau) = e_{t4}(\tau) \oplus \mathrm{FTTC}(t_4) = 1[3,\ 5.5,\ 9,\ 11.5]$, and we get $\pi_{p5}(\tau) = 1[4,\ 7.5,\ 12,\ 15.5]$, successively $e_{t6}(\tau) = 1[4,\ 7.5,\ 12,\ 15.5]$, $o_{t6}(\tau) = 1[4,\ 7.5,\ 12,\ 15.5]$, $\pi_{p7}\ (\tau) = 1[12,\ 17,\ 23,\ 28]$.

Step 4: p_2, t_3, p_4, t_5 and p_6. $e_{t3}(\tau) = \pi_{p2}(\tau) \oplus \mathrm{FTPC}(p_2) = 1[2,\ 3,\ 5,\ 6]$, so we get $o_{t3}(\tau) = e_{t3}(\tau) \oplus \mathrm{FTTC}(t_3) = 1[2,\ 3,\ 5,\ 6]$ and $\pi_{p4}(\tau) = o_{t3}(\tau) \oplus d_{t3}(\tau) =$

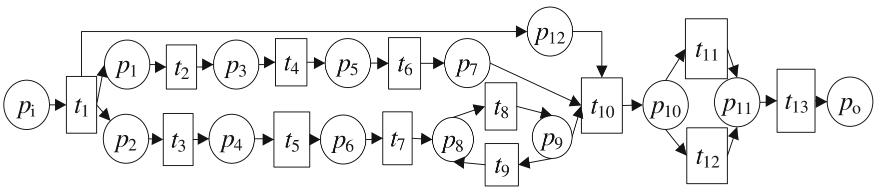


Fig. 9 AFTWF-nets model of business process

Table 1 Significances and time information of the transtions and places

t	Significance	*FTTC(t)*	*FD(t)*	p	Significance	*FTPC(p)*
t_1	Assignment	1[0, 0, 0, 0]	1[2, 3, 5,6]	p_i	Process begins	1[0 ,0, 0, 0]
t_2	Apply for funds	1[0, 0, 0, 0]	1[1, 2, 3, 4]	p_o	Process ends	1[0, 0, 0, 0]
t_3	Apply for funds	1[0, 0, 0, 0]	1[0, 1, 2, 3]	p_1	Local purchasing Dept	1[0, 0, 0, 0]
t_4	Appropriation	1[0, 0.5, 1, 1.5]	1[1, 2, 3, 4]	p_2	Subsidiary purchasing Dept	1[0, 0, 0, 0]
t_5	Appropriation	1[0, 0, 0, 0]	1[1, 2, 3, 4]	p_3	Local finance office	1[0, 0, 0, 0]
t_6	Purchase& deliver	1[0, 0, 0, 0]	1[8, 9.5, 11, 12.5]	p_4	Subsidiary finance office	1[0, 0, 0, 0]
t_7	Purchase& deliver	1[0, 0, 0, 0]	1[7, 8, 9, 10]	p_5	Funds	1[0, 0, 0, 0]
t_8	Quality check	1[0, 0, 0, 0]	1[1, 2, 3, 4]	p_6	Funds	1[0, 0, 0, 0]
t_9	Recheck	1[0, 0, 0, 0]	1[0, 0, 1, 1]	p_7	Raw material	1[0, 0, 0, 0]
t_{10}	Deliver production	1[0, 0, 0, 0]	1[0, 1, 1, 2]	p_8	Raw material	1[0, 0, 0, 0]
t_{11}	Produced by line A	1[1, 2, 2, 3]	1[8, 10, 12, 14]	p_9	Quality check Dept	1[0, 0, 0, 0]
t_{12}	Produced by line B	1[2, 3, 3, 4]	1[5, 6, 7, 8]	p_{10}	Production Dept	1[0, 0, 0, 0]
t_{13}	Deliver to users	1[0, 0, 0, 0]	1[0, 1, 2, 3]	p_{11}	Productions	1[0, 0,0, 0]
				p_{12}	Warehouse	1[0, 5, 25, 30]

$1[2, 4, 7, 9]$, and $e_{t5}(\tau) = 1[2, 4, 7, 9], o_{t5}(\tau) = 1[2, 4, 7, 9]$, $\pi_{p6}(\tau) = 1[3, 6, 10, 13]$.

Step 5: t_7, p_8, t_8, p_9 and t_9. Before loop $\pi_{p6}(\tau) = 1[3, 6, 10, 13], e_{t7}(\tau) = 1[3, 6, 10, 13]$ $o_{t7}(\tau) = 1[3, 6, 10, 13], e_{t8}(\tau) = 1[10, 14, 19, 23], o_{t8}(\tau) = 1[10, 14, 19, 23]$ and $\pi_{p9}(\tau) = 1[11, 16, 22, 27]$. Assume possibility that raw materials do not meet the standard, is 10 %, and when they are unqualified, t_9 is enabled, and then $e_{t9}(\tau) = 1[11, 16, 22, 27]$, $o_{t9}(\tau) = 1[11, 16, 22, 27]$, $d_{t9}(\tau) = 1[0, 0, 1, 1]$, so $\pi_{p8}(\tau)' = 1[11, 16, 23, 28]$, $o_{t8}(\tau)' = 1[11, 16, 23, 28]$, and $\pi_{p9}(\tau)' = 1[12, 18, ;26, 32]$. If circling once, $\pi_{p9}(\tau)'' = 90\ \% \times \pi_{p9}(\tau) + 10\ \% \times \pi_{p9}(\tau)' = 1[11.1, 16.2, 22.4, 27.5]$.

Step 6: t_{10} $e_{t10}(\tau) = \text{latest}(\pi_{p7}(\tau), \pi_{p9}(\tau)'', \pi_{p12}(\tau)) = \min(1, 1, 1)[\max(12, 11.1, 0), \max(17, 16.2, 5), \max(23, 22.4, 25), \max(28, 27.4, 30)] = 1[12, 17, 25, 30]$, so $o_{t10}(\tau) = 1[12, 17, 25, 30]$, $\pi_{p10}(\tau) = 1[12, 18, 26, 32]$.

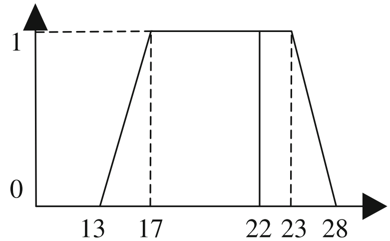
Step 7: $t_{11}$and $t_{12}.e_{t11}(\tau) = e_{t12}(\tau) = 1[12, 18, 26, 32]$, so $o_{t11}(\tau) = \text{MIN}\{1[12, 18, 26, 32] \oplus 1[1, 2, 2, 3],\ \text{earliest}(1[12, 18, 26, 32] \oplus 1[1, 2, 2, 3], 1[12, 18, 26, 32] \oplus 1[2, 3, 3, 4])\} = \min\{1[13, 20, 28, 35], 1[13, 20, 28, 35]\}= 1[13, 20, 28, 35]$, $o_{t12}(\tau) = \text{MIN}\{1[12, 18, 26, 32] \oplus 1[2, 3, 3, 4], \text{earliest}(1[12, 18, 26, 32] \oplus 1[1, 2, 2, 3], 1[12, 18, 26, 32] \oplus 1[2, 3, 3, 4])\} = \min\{1[14, 21, 29, 36], 1[13, 20, 28, 35]\} = 1[14, 21, 28, 35], so if execute $t_{I1,}$ $\pi_{p11}(\tau) = o_{t11}(\tau) \oplus d_{t11}(\tau) = 1[21, 30, 40, 49]$, and if execute $t_{I2,}$ $\pi_{p11}(\tau)' = o_{t12}(\tau) \oplus d_{t12}(\tau) = 1[18, 26, 35, 43]$, and assume possibility that execute t_{I1} is 60 %, that execute t_{I2} is 40 %, so $\pi_{p11}(\tau)'' = 60\ \% \times \pi_{p11}(\tau) + 40\ \% \times \pi_{p11}(\tau)' = 1[19.8, 28.4, 38, 46.6]$.

Step 8: t_{13} and $p_o.e_{t13}(\tau) = 1[19.8, 28.4, 38, 46.6]$, $o_{t13}(\tau) = 1[19.8, 28.4, 38, 46.6]$ and $\pi_{po}(\tau) = 1[19.8, 29.4, 40, 49.6]$. It means that the total time cost is $1[19.8, 29.4, 40, 49.6]$.

After calculation, time performance of the workflow is shown as follows:

1. Activities execution time constraint satisfaction. For example, it is expected that local purchasing Dept should finish purchasing within 22 days, which means the token arrives at p_7 at $\pi_{p7}(\tau) = 1[22, 22, 22, 22]$. As shown in Fig. 10, the area of the left part of the trapezoidal is [(22 − 17) + (22 − 13)] × 1/2 = 7, and the whole area is [(23 − 17) + (28 − 13)] × 1/2 = 10.5, so the possibility that t_6 finishes within 22 days is Possibility$(\pi_{p7}(\tau) \leq \pi(\tau)) = 7/10.5 = 0.667$.

Fig. 10 $\pi_{p7}(\tau) \leq \pi(\tau)$



2. Time distance activities constraints satisfiability. There are some time constraints between production Dept receiving raw materials and delivering product to users. If time distance between t_{10} and t_{13} needs to be less than 5 days, it means to solve Possibility$(o_{t13}(\tau) \leq o_{t10}(\tau) \oplus 1[5, 5, 5, 5])$. As shown in Fig. 11,

Step 1:

$$\begin{aligned}&\text{Possibility}(o_{t13}(\tau) \leq o_{t10}(\tau) \oplus 1[5, 5, 5, 5])\\&\quad = \text{Possibility}(1[19.8, 28.4, 38, 46.6] \leq 1[17, 22, 30, 35]).\end{aligned}$$

Fig. 11 $o_{t13}(\tau) \leq o_{t10}(\tau) \oplus 1[5, 5, 5, 5]$

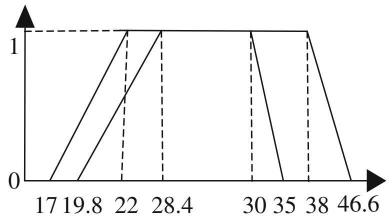
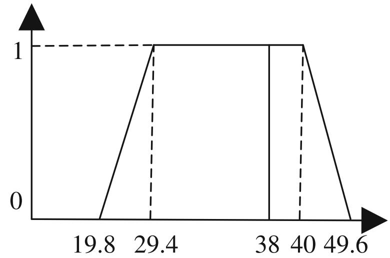


Fig. 12 $\pi_{\text{po}}(\tau) \leq \pi_f(\tau)$



Step 2: The overlap area is [(30 − 28.4) + (35 − 19.8)] × 1/2 = 8.4, the area of right half part is [(38 − 28.4) + (46.6 − 19.8)] × 1/2 = 18.2,so $\text{Possibility}(o_{t13}(\tau) \leq o_{t10}(\tau) \oplus 1[5,\ 5,\ 5,\ 5]) = 8.4/18.2 = 0.462$,

Step 3: That is the possibility that time distance is no more than 5 days.

3. Temporal reachable probability of the process. Firstly, calculate it using general method. From the above result, the total fuzzy time cost is $\pi_{\text{po}}(\tau) - \pi_{\text{pi}}(\tau) = 1[19.8,\ 29.4,\ 40,\ 49.6]$.

If the whole process finishes within 38 days, the expected fuzzy timestamp is $\pi_f(\tau) = 1[38,\ 38,\ 38,\ 38]$. As shown in Fig. 12, the area of the whole trapezoidal is [(40 − 29.4) + (49.6 − 19.8)] × 1/2 = 20.2, the left part is [(38 − 29.4) + (38 − 19.8)] × 1/2 = 13.4, so $\text{Possibility}(\pi_{po}(\tau) \leq \pi_f(\tau)) = 13.4/20.2 = 0.663$. Secondly, calculate it using stratification algorithm. HA-FTWF-nets is shown in Fig. 13 and the time information of each node is given above; therefore, we can obtain the time information of each basic route in level 3, substitute the result into the next upper level until finally substitute the result into level 0 and work out $\pi_{\text{po}}(\tau) = 1[19.8,\ 29.4,\ 40,\ 49.6]$, which is the same as the above answer.

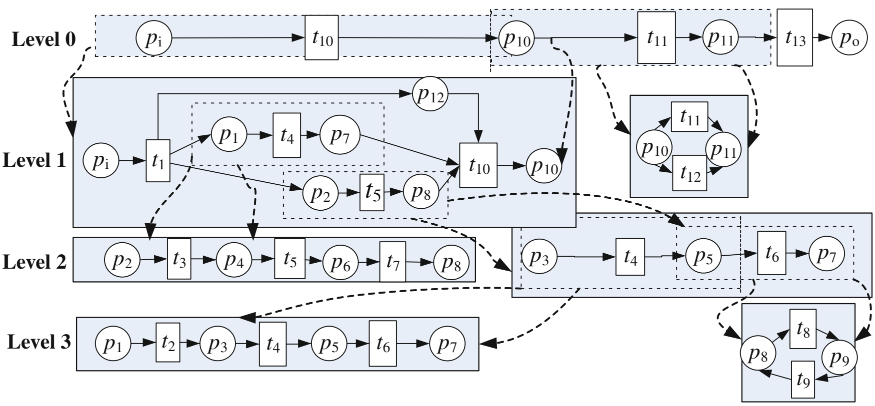


Fig. 13 HAFWF-nets model of business process

5 Conclusion

This paper proposed AFTWF-nets based on the existing research achievements on TCPN and FTHN and then gave its formal definition as well as corresponding reasoning and analysis algorithm. Firstly, AFTWF-nets has been used to describe the temporal information within workflows, and then model time constraints, and analyze its time performance. Secondly, based on the above analysis, a reduction algorithm for fuzzy time workflow model was put forward. Finally, through a practical interprovincial company application, the effectiveness of our method has been verified.

AFTWF-nets can be used to model those workflows which involved uncertain time information, and it will increase the flexibility of the workflow management systems, and also enrich workflow modeling theory and, promote the application of workflow management software.

References

1. Li H, Fan Y (2002) Overview on managing time in workflow systems. J J Softw 13(8):1552–1558
2. vander Aalst WMP (1998) The application of Petri nets to workflow management. J J Circuits Syst Comput 8(1):21–661
3. Ling S, Schmidt H (2000) Time Petri nets for workflow modeling and analysis. In: Proceedings of IEEE international conference on systems, Man and Cybernetics, IEEE Press, Nashville, pp 3039–3041
4. Li H, Fan Y (2004) Workflow model analysis based on time constraint Petri nets. J J Softw. 15(1):17–26
5. Murata T (1996) Temporal Uncertainty and fuzzy-timing high-level Petri nets. In: Application and theory of Petri Nets (Lecture Notes in Computer Science), New York, pp 11–28
6. Pan Y, Tang Y (2006) Modeling of fuzzy temporal workflow nets and its time possibility analysis. J Comput Integr Manuf Syst. 12(11):1779–1784
7. Liu X, Yin G, Zhang Z (2009) A high-level fuzzy timed Petri-net-based approach. In: 2009 Fourth international conference on internet computing for science and engineering, pp 131–136