



A Decomposition Algorithm of Petri Net Utilizing Index Function

Kai-Qing Zhou^a, Li-Ping Mo^a, Chang-Feng Chen^a, Wei Jiang^a

^aCollege of Information Science and Engineering, Jishou University, Jishou, 416000, Hunan, China

Abstract. Recently, it is difficult to simulate, analyze and control a real knowledge-based system using the correspondence Petri net (PN) when there exist many current states. To overcome the state explosion problem of PN, an efficient decomposition algorithm is presented to divide a large-scale PN into a series of corresponding sub-PNs by keeping the consistency of dynamic properties. In this novel decomposition approach, an index function is defined to judge the subnet needs to be decomposed or not. Furthermore, an exhaustive analysis on the consistency of related dynamic properties is also discussed between the original PN and the corresponding sub-PNs. Finally, a case study is carried out to illustrate the feasibility and validity of the proposed approach.

1. Introduction

SINCE Petri net (PN) was proposed by C. A. Petri in 1962, an acknowledged shortage of PN, namely state space explosion, has restrained the further development of PN and its application [1]. Until now, this issue has not been completely resolved. Normally, the PN with limited nodes can be used to implement simulation, analysis and verification process [2]. However, the operation of simulating, analyzing and verifying large-scale PN is increasingly difficult due to the state space explosion [3]. To overcome these problems, the idea of simplification technique of PN was proposed to reduce the scale of corresponding PN by using equivalent transformation method. In other words, the technique is to divide large-scale and complexity PN into a group of small-scale, easy-to-analyze sub-PNs which have the same properties as original [4-6]. According to the decomposition thinking, various decomposition approaches were employed to control the PN scale in different industrial areas. Liu et al. [7] proposed a decomposition method to divide a PN model of concurrent programs into multiple process nets for avoiding the deadlocks by using the number of processes and message places in a concurrent program. Focused on the state explosion issue of business process, Wang et al. [8] proposed a new mining technique to increase the availability of enterprise information system by using a workflow decomposition algorithm based on PN theory. The increasing volume of data impacts the further studies of process mining area from two aspects both of opportunities and challenges, Aalst [9] carried out a generic decomposition algorithm by using PN to simply the mining

2010 *Mathematics Subject Classification.* Primary 68Q85; Secondary 68Q05

Keywords. Decomposition algorithm, Dynamic properties, Index function, Petri net, State explosion

Received: 19 October 2018; Revised: 19 November 2018; Accepted: 21 December 2018

Communicated by Bolin Liao

This work is supported by the Research Foundation of Education Bureau of Hunan Province (No.18B317), National Natural Science Foundation of China (Nos. 61741205 and 61966014), and Hunan Provincial Natural Science Foundation of China (No. 2019JJ40234)

Email addresses: kqzhou@jsu.edu.cn (Kai-Qing Zhou), zmx89@jsu.edu.cn (Li-Ping Mo), 740839301@qq.com (Chang-Feng Chen), jeanwaymiss@163.com (Wei Jiang)

process, which could be combined with different existing process discovery and conformance checking techniques. Zhou et al. [10] proposed to decompose a large scale fuzzy PN (FPN) model into a series of sub-FPN models with completed inner-inference-paths for making the FPN model more adaptive in the complex engineering applications. Furthermore, Zhou et al. [11] presented a bi-directional reasoning algorithm using FPN formalism to execute fault diagnosis function for a complex manufacturing system by removing the unrelated places and transitions of the goal place. Li et al. [12] used a group of simple algebraic operations to decompose PN model and employed the proposed algorithm to model and analyze the mail-sorting systems. Chen et al. [13] utilized a matching theory to divide a large-scale PN model. Nishi and Matsumoto [14] apply the PN decomposition algorithm to gain an optimal solution of deadlock-free and non-cyclic scheduling of dual-armed cluster tools for decreasing the computational complexity of the semiconductor cluster tool system. Dideban and Zeraatkar [15] proposed a decomposition algorithm of PN to reduce the cost of controller synthesis operation in discrete event systems utilizing the P-invariant property. Ye et al. [16] presented a two-phase decomposition algorithm to divide a large-size controller. Other typical applications of the decomposition algorithm using PN in industrial fields can be found in references [17-20]. Although various decomposition algorithms have been proposed to overcome the state explosion issue of PN and gained fruitful results, the existing literature did not analyze the proposed algorithms own the ability to keep the consistency of dynamic properties between the original PN and the obtained sub-PN models. From this viewpoint, a decomposition algorithm utilizing index function is proposed in this paper to decompose a large-scale PN into a series of corresponding sub-PNs via keeping the consistency of the dynamic properties rooted in our previous works [21-22]. The main contributions of this article could be classified into the following three aspects.

(1) Propose a decomposition algorithm for separating a large-scale PN into a series of sub-PNs utilizing a defined index function.

(2) Investigate the consistency of the dynamic properties including dynamic place, liveness, boundedness & safeness, and fairness between the original PN model and the gained corresponding decomposed sub-PN models. Moreover, some theorems are also proposed based on the discussions above.

(3) Illustrate the validity and practicability of this method through theoretical analysis and apply this algorithm into a case study.

The remaining sections are organized as follows. Section 2 explains the related information on PN. Section 3 discusses the proposed algorithm and gives an example to illustrate its implementation process of our algorithm. In Section 4, consistency on dynamic properties between the original PN and decomposed sub-nets is discussed. In Section 5, a case study is given to demonstrate the solution of the algorithm, and Section 6 presents conclusion and future work.

2. Petri Net and Relevant Information

The related notions, dynamic properties and analysis methods of PN are introduced based on previous literature mentioned above.

A. Related Notions

Definition 1 Petri Net: Petri Net (PN) is defined as six-tuple: $\Sigma = \{P, T; F, K, W, M_0\}$. Where, P is a finite set of places, T is a finite set of transitions, $F \subseteq (P \times T) \cup (T \times P)$ is a finite set of arcs, $K = \{1, 2, 3, \dots\}$ is a capacity function of p , $W : F \rightarrow \{1, 2, 3, \dots\}$ is a weight function, and M_0 is the initial marking.

Definition 2 Pre-set and Post-set: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, $\bullet x = \{y | (y, x) \in F\}$ the pre-set of x and $x^\bullet = \{y | (x, y) \in F\}$ is the post-set or output set of x . Where, $x, y \in P \cup T$.

Definition 3 Enabling Rule: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, a transition t is enabled when the marking $M(p)$ can be fired by

$$\forall p \in \bullet t : M(p) \geq W(p, t) \wedge \forall p \in t^\bullet : M(p) + W(t, p) \leq K(p) \quad (1)$$

(1) M' is M 's succeeding fact after t is enabled, marked as $M[t > M'$

(2) Result of transition is enabled

If there exists a $M[t > \cdot]$, the transition t can be enabled under M , and $M[t > M']$. The result of M' is described as follows.

For $\forall p \in P$

$$M'(p) = \begin{cases} M(p) - W(p, t) & p \in \bullet t - t^\bullet \\ M(p) + W(t, p) & p \in t^\bullet - \bullet t \\ M(p) - W(p, t) + W(t, p) & p \in \bullet t \cap t^\bullet \\ M(p) & p \notin \bullet t^\bullet \end{cases} \quad (2)$$

B. Dynamic Properties

The properties are used to explore the question such as ‘what can we do with the PN models?’. The properties could be classified into two types: dynamic properties (depend on the initial marking) and structure properties (independent of the initial marking). The discussion is carried out on the dynamic properties, which are reachability, boundedness and safeness, liveness, and fairness.

Properties 1: Reachability

Reachability is the basis to explore the dynamic properties of any system. The related concepts are described as follows.

Definition 4: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, the direct reachable condition from M to M' is that there exist $t \in T$, transition sequence t_1, t_2, \dots, t_k and marking sequence M_1, M_2, \dots, M_k , then $M[t_1 > M_1[t_2 > M_2 \dots M_{k-1}[t_k > M_k]$ $R(M)$ is a set of all markings from M .

Definition 5: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, M_0 is the initial marking. $R(M_0)$ is the reachability marking set which is the smallest set and meet two conditions, which are

- (1) $M_0 \in R(M_0)$;
- (2) $M \in R(M_0)$ and there exists a $t \in T$ such as $M[t > M']$ and $M' \in R(M_0)$.

Properties 2: Boundedness and Safeness

Definition 6: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, the sufficient condition of place P is bounded and could be described as

For $p \in P$, if there exists a positive integral B such as $\forall M \in R(M_0): M(p) \leq B$. The bound of place p marked as $B(p)$ is the smallest positive integral:

$$B(p) = \min \{B | \forall M \in R(M_0) : M(p) \leq B\} \quad (3)$$

p is safe iff $B(p) = 1$

Definition 7: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, Σ is bounded when any $p \in P$ is bounded, and the bounded of Σ is $B(\Sigma) = \max \{B(p) | p \in P\}$ $\Sigma = \{P, T; F, K, W, M_0\}$, Σ is safe iff $B(\Sigma) = 1$

Properties 3: Liveness

Definition 8: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, M_0 is an initial marking and $t \in T$. The condition of transition t is live could be described as for any $M_0 \in R(M_0)$, there exists $M' \in R(M_0)$ such as $M[t > \cdot]$. Σ is live means that any $t \in T$ is live.

However, the definition 8 is too strict and only limited number of PN model owns this property. For this reason, different levels liveness are introduced as definition 9.

Definition 9: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, $t \in T$.

- (1) Level-0 live (or dead): if t can never be fired in any firing sequence.
- (2) Level-1 live: $\exists M \in R(M_0) : M[t > \cdot]$
- (3) Level-2 live: For any integral n , there exists $\sigma \in T^*$ such as $M_0[\sigma > \cdot]$ and $\#(t/\sigma) \geq n$ Where, $\#(t/\sigma) \geq n$ represents the appearing number of t in sequence.
- (4) Level-3 live: If there exists an infinite transition sequence σ such as that of $M_0[\sigma > \cdot]$ and the times that t appears in σ is infinite.
- (5) Level-4 live (or live): If for any $M \in R(M_0)$, t is the level-1 live in σ

Properties 4: Fairness

Fairness discusses the relationships between two transitions of two transition groups.

Definition 10: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, $t_1, t_2 \in T$. The condition of t_1 and t_2 belongs to the fair relation and is given below.

If there exists a positive integral k , for any $M \in R(M_0)$ and any $\sigma \in T^* : M[\sigma >$ such as following equation.

$$\#(q_{ij}/\sigma) = 0 \rightarrow \#(q_{ij}/\sigma) \leq ki, j \in \{1, 2\}, i \neq j \tag{4}$$

Σ is a fair PN model when any two transitions in Σ belongs to fair relation.

C. Incidence Matrix and State Equation

par Incidence matrix and state equation are widely applied into PN area to analyze the relative properties of PN. The corresponding notions are introduced below, respectively.

Definition 13: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$ (n transitions and m places), the incidence matrix $C = [c_{ij}]_{m \times n}$ is $C_{ij} = C_{ij}^+ - C_{ij}^-$, where:

$$C_{ij}^+ = \begin{cases} 1 & (t_j, p_i) \in F \\ 0 & \text{others} \end{cases} \quad \text{and} \quad C_{ij}^- = \begin{cases} 1 & (p_i, t_j) \in F \\ 0 & \text{others} \end{cases} \quad (i = \{1, 2, \dots, m\}, j = \{1, 2, \dots, n\}) \tag{5}$$

Definition 14: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, M_0 is the initial marking of Σ , C is the incidence matrix of Σ . If there are $M \in R(M_0)$ and an nonnegative integer vector X , the state equation is described as $M = M_0 + CX$ and an nonnegative integer vector X , the state equation is described as $M = M_0 + CX$.

3. A Decomposition Algorithm of PN model

The main idea of the proposed algorithm is that equivalent transforms the original PN to a group of subnets using an index function. The index function for the related notions is then introduced.

A. Index Function and Decomposition

Definition 15: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, function $f : P \rightarrow \{1, 2, \dots, k\}$ and $\forall p_1, p_2 \in P, \exists t \in T$. If there exists $\{p_1, p_2\} \subseteq t^\bullet \vee \{p_1, p_2\} \subseteq \bullet t$ such that $f(p_1) \neq f(p_2)$. Then, f is the index function of Σ and $f(p)$ is the index of place p .

Definition 16: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, $f : P \rightarrow \{1, 2, \dots, k\}$ is the index function of Σ . $\Sigma_i = (P_i, T_i; F_i, M_{0_i})$ ($i = 1, 2, \dots, k$) are the corresponding decomposed subnets of Σ , where:

- (1) $P_i = \{p \in P | f(p) = i\}, i = 1, 2, \dots, k$
- (2) $T_i = \{t \in T | \exists p \in P, t \in p^\bullet \cup \bullet p\}, i = 1, 2, \dots, k$
- (3) $F_i = F \cap \{(P_i \times T_i) \cup (T_i \times P_i)\}, i = 1, 2, \dots, k$
- (4) $M_{0_i} = R(M_0), i = 1, 2, \dots, k$

Based on the definitions 15 and 16, a $\lambda(p)$ is proposed in definition 17 to represent the number of place which belongs to the same pre-set or post-set of transition in place set.

Definition 17: For $\forall p \in P$,

$$\lambda(p) = \left| \text{num}(p_1) | \forall t \in T, \exists p_1 \in P, p_1 \neq p, \text{s.t.} \begin{cases} \text{if } p \in t^\bullet, p_1 \in t^\bullet \\ \text{if } p \in \bullet t, p_1 \in \bullet t \end{cases} \right|$$

Where $\text{num}(p)$ is the number of satisfied requirement places.

B. The Proposed Decomposition Algorithm

The steps of this algorithm are given as follows.

Input: PN model $\Sigma = (S, T; F)$.

Output: The index function value of each place p_i .

Step 1: $X = P, Y = \emptyset, k = 1$;

Step 2: $X = \emptyset$, move to step 6. Otherwise, calculate each place's $\lambda(p)$ in X ;

Step 3: If there are some $\lambda(p)$ in X which is not zero, then choose a place $\lambda(p) \neq 0$ and move to a new set. Meanwhile, delete the place and related arcs in original net. The new net expressed as $N' = (P', T; F')$. If $X = P'$, move to Step 2; Otherwise, move to step 4;

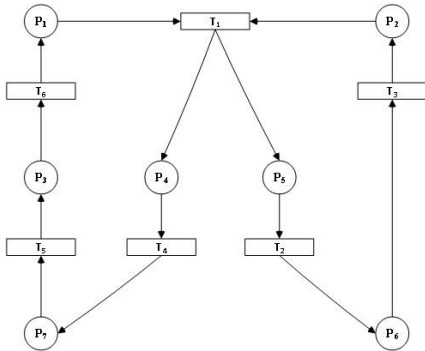


Fig. 1. Original PN Model

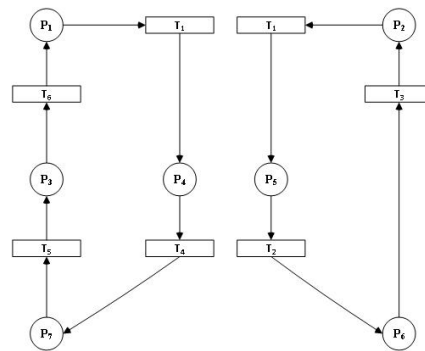


Fig. 2. Two decomposed subnets

Step 4: If all places in X can be connected, then $X_k = X$, $X = Y$, $Y = \emptyset$, $k = k + 1$; and back to step 2; Otherwise, move the unconnected places in X to Y , and X, Y express as X', Y' . $X_k = X'$, $X = Y'$, $Y = \emptyset$, $k = k + 1$, move to step 2;

Step 5: For $\forall p \in P$, if $p \in X_k$, $f(p) = k$.

C. Algorithm Complexity

The core operation of the presented is the step 3. The step 3 could be divided into two phases. Assume there are n places in the vector X . The function of the 1st phase is to check $\lambda(p)$ of each place p is equal to 0 in X one-by-one. The complexity of the 1st phase is n . In the 2nd phase, if there exists $\lambda(p)$ is not 0, choose any one place ($\lambda(p) \neq 0$) from the X and move it to a new set. Then, define the new net as $N' = (P', T; F')$. Otherwise, move to step 4. At this time. The complexity of the 2nd phase is also n . Hence, the algorithm complexity of the step 3 is n^2 . Furthermore, the entire algorithm complexity of the presented algorithm is n^2 .

D. Simple Example of Implementing the Proposed Algorithm

A PN could be divided into a series of simple subnets after implementing the proposed algorithm. An example is used to illustrate the proposed algorithm. Assume the original PN model is described in Figure 1 and use $p(k)$ to represent $\lambda(p) = k$. The implementation process is given in the following steps.

Step 1: $X = P$, $Y = \emptyset$, $k = 1$;

Step 2: Calculate $\lambda(p)$ for $X = \{p_1(1), p_2(1), p_3(0), p_4(1), p_5(1), p_6(0), p_7(0)\}$.

Step 3: Move p_1 ($\lambda(p_1) \neq 0$) to Y , then $X = \{p_2(0), p_3(0), p_4(1), p_5(1), p_6(0), p_7(0)\}$, $Y = \{p_1\}$;

Step 4: Move p_4 to Y , then $X = \{p_2(0), p_3(0), p_5(0), p_6(0), p_7(0)\}$, $Y = \{p_1, p_4\}$;

Step 5: p_3 and p_7 in X are unconnected with other places, so move them to Y .

$X = \{p_2(0), p_5(0), p_6(0)\}$, $Y = \{p_1, p_3, p_4, p_7\}$

Step 6: Because the $\lambda(p)$ in X both equals zero. $X_1 = \{p_2, p_5, p_6\}$, $X = \{p_1, p_3, p_4, p_7\}$, $Y = \emptyset$, $k = 2$;

Step 7: The results are $f(p_1) = f(p_3) = f(p_4) = f(p_7) = 2$ and $f(p_2) = f(p_5) = f(p_6) = 1$

Finally, the PN is decomposed into two corresponding subnets as Figure 2.

4. Analysis on the consistency of dynamic properties

The proposed algorithm is verified by the analysis of consistency.

a. Dynamic place

The consistency of the dynamic place means that the number of token in decomposed subnets can keep unanimous with the original PN model after being enabled by the same transition sequence.

Theorem 1: For a $\Sigma = \{P, T; F, K, W, M_0\}$, $\Sigma_i = \{P_i, T_i; F_i, K_i, W_i, M_{0_i}\} (i = 1, 2, \dots, k)$ is the corresponding subnet. If there exists $M_0 \geq \sum M_i \in M_{0_i} (i = 1, 2, \dots, k)$ such as $M_i(p_i) = M(p_i)$ after being enabled by the same transition sequence.

Proof: To prove this theorem, some hypotheses are given as follows.

(1) The Σ 's incidence matrix is C , the subnet Σ_i 's incidence matrix is C_i . C_i has the same number of row and column as C , and the corresponded place (or transition) in the same row (or column) in C_i is the same with that of C ;

(2) In incidence matrix, zero represents that the transition or place does not appear in Σ_i , u, u_i ($i = 1, 2, \dots, k$) represent the transition sequence in original PN and subnets (same dimension), and zero means the transition in u_i which does not appear;

(3) The initial marking $M_0, M_0, (i = 1, 2, \dots, k)$ has the same dimension, the value of place which does not appear in M_0 is zero.

The PN is divided into two parts, simple subnet and complexity subnet. The complexity subnet is implemented by the decomposition strategy. It is done step by step, until the transition's pre-set and post-set belong to different subnets respectively.

Then, we can simplify the number of subnets to two subnets and generate the incidence matrices as follows.

$$C_1 = \begin{matrix} & T_1 \cap T_2 & T_1 - T_2 & T_2 - T_1 \\ \begin{matrix} p_1 \\ p_2 \end{matrix} & \begin{pmatrix} t_0 & t_0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{matrix} \quad C_2 = \begin{matrix} & T_1 \cap T_2 & T_1 - T_2 & T_2 - T_1 \\ \begin{matrix} p_1 \\ p_2 \end{matrix} & \begin{pmatrix} 0 & 0 & 0 \\ t_1 & 0 & t_1 \end{pmatrix} \end{matrix} \quad C = \begin{matrix} & T_1 \cap T_2 & T_1 - T_2 & T_2 - T_1 \\ \begin{matrix} p_1 \\ p_2 \end{matrix} & \begin{pmatrix} t_0 & t_0 & 0 \\ t_1 & 0 & t_1 \end{pmatrix} \end{matrix} \quad (6)$$

Then, Σ and Σ_i have the same transition sequence.

$$\begin{aligned} u_1 &= \begin{pmatrix} T_1 \cap T_2 & T_1 - T_2 & T_2 - T_1 \\ Z_0 & Z_1 & 0 \end{pmatrix}^T \\ u_2 &= \begin{pmatrix} T_1 \cap T_2 & T_1 - T_2 & T_2 - T_1 \\ Z_0 & 0 & Z_2 \end{pmatrix}^T \\ u &= \begin{pmatrix} T_1 \cap T_2 & T_1 - T_2 & T_2 - T_1 \\ Z_0 & Z_1 & Z_2 \end{pmatrix}^T \end{aligned} \quad (7)$$

There exists an equation as follows.

$$M_{0_1} = \begin{pmatrix} p_1 & p_2 \\ k_0 & 0 \end{pmatrix}^T \quad M_{0_2} = \begin{pmatrix} p_1 & p_2 \\ 0 & k_1 \end{pmatrix}^T \quad M_0 = \begin{pmatrix} p_1 & p_2 \\ k_0 & k_1 \end{pmatrix}^T \quad (8)$$

Finally, the results are as follows.

$$M_1 = M_{0_1} + C_1 u_1 = \begin{pmatrix} p_1 & p_2 \\ (k_0 + (z_0 + z_1)t_0, 0) \end{pmatrix}^T \quad (9)$$

$$M_2 = M_{0_2} + C_2 u_2 = \begin{pmatrix} p_1 & p_2 \\ (0, k_1 + (z_0 + z_2)t_1) \end{pmatrix}^T \quad (10)$$

$$M = M_0 + C u = \begin{pmatrix} p_1 & p_2 \\ (k_0 + (z_0 + z_1)t_0, k_1 + (z_0 + z_2)t_1) \end{pmatrix}^T \quad (11)$$

Hence, the theorem 1 is proved and $M_i(p_i) = M$ ($p_i \in P$)

Theorem 2: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, $\Sigma_i = \{P_i, T_i; F_i, K_i, W_i, M_{0_i}\} (i = 1, 2, \dots, k)$ is the corresponding subnet. The necessary and sufficient condition of Σ is live is that Σ_i is live.

Proof: (Necessity) If Σ is live \Leftrightarrow Each $t \in T$ in Σ is live \Leftrightarrow For $t \in T$, any reachability marking $M \in [M_0 >$ has $M' \in [M >$ from $M, M' [>$ holds.

Assume at least one of Σ_i is dead and assume Σ_p is the decomposed subnet. It means that there is at least one transition in Σ_i is dead. This is in contrast to the previous conclusion All $t \in T$ in Σ are live. So the assumption does not hold water. The necessity is proved.

(Sufficiency) If each Σ_i is live. Conclusions can get about Σ as follows.

$$P = \bigcup_{i=1}^k P_i \text{ where, } P_i \cap P_j = \emptyset \ (i, j = 1, 2, \dots, k, i \neq j)$$

$$T = \bigcup_{i=1}^k T_i \text{ where, } T_i \cap T_j \neq \emptyset \ (i, j = 1, 2, \dots, k, i \neq j)$$

$$F = \bigcup_{i=1}^k F_i$$

$$M_0(p) = \begin{cases} M_{0_1} & (p \in P_1) \\ M_{0_2} & (p \in P_2) \\ \dots & \\ M_{0_k} & (p \in P_k) \end{cases}$$

Assume Σ is dead \Leftrightarrow At least one of $t \in T$ in Σ is dead. $\Leftrightarrow T = \bigcup_{i=1}^k T_i$, t belongs to one of T_i . Assume t belongs to T_p , and then T_p is dead. This is in contrast to the previous conclusion. The assumption does not hold. The necessity is proved.

Lemma 1[23] For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, and C is the corresponding incidence matrix of Σ . The necessary and sufficient condition Σ of is bounded is that live is that $\exists m$ dimensional positive integral vector Y , s.t. $CY \leq 0$, where $m = |P|$ means the number of place in Σ .

Theorem 3: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, $\Sigma_i = \{P_i, T_i; F_i, K_i, W_i, M_{0_i}\} (i = 1, 2, \dots, k)$ is the subnet. The necessary and sufficient condition of Σ is bounded is that Σ_i is bounded.

Proof: (Necessity) If Σ is bounded \Leftrightarrow There exists a conclusion as follows.

For $p \in P$ in Σ , \exists positive integral B , $\forall M \in R(M_0)$ then $M(p) \leq B$ holds.

Based on theorem 1, for $\forall M \in R(M_0)$, the token is consistent in Σ and Σ_i . For $p_i \in P_i$, there exists a positive integral B_i such as $M_i(p_i) \leq B_i$ by $\forall M_i \in R(M_{0_i})$, where B_i and B 's relation can be described as

$$B = \min_{i=1}^k \{B_i | \forall M_i \in R(M_{0_i}) : M_i(p_i) \leq B_i\} \tag{12}$$

(Sufficiency) If Σ_i is bounded, assume the incidence matrix of Σ_i is C_i . From the Lemma 1, there exists a m_i dimensional positive integral vector $Y_i (i = 1, 2, \dots, k)$ such as that of $C_i Y_i \leq 0 (i = 1, 2, \dots, k)$, where $m_i = |P_i| (i = 1, 2, \dots, k)$.

There exists $C = (C_1, C_2, \dots, C_k)$ which satisfies the following equations.

$$(C_1, C_2, C_3, \dots, C_k) \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_k \end{pmatrix} = C_1 Y_1 + C_2 Y_2 + C_3 Y_3 \dots + C_k Y_k \leq 0 \tag{13}$$

There exists a m dimensional positive integral vector $Y = (Y_1, Y_2, Y_3, \dots, Y_k)^{-1}$ such as $CY \leq 0$. Where,

$$m = \sum_{i=1}^k m_i = \sum_{i=1}^k |P_i| \text{ and } P = \bigcup_{i=1}^k P_i$$

Furthermore, $p_i \cap p_j = \emptyset (i, j = 1, 2, \dots, k, i \neq j)$.

So, there is $\sum_{i=1}^k |P_i| = |P|$. So $m = |P|$ and Σ is bounded.

Theorem 4: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, $\Sigma_i = \{P_i, T_i; F_i, K_i, W_i, M_{0_i}\} (i = 1, 2, \dots, k)$ s the subnet. The necessary and sufficient condition of Σ is safe is that each Σ_i is safe.

Proof: Based on theorem 3, for $t \in T$, there is $B(p) = \min_{i=1}^k \{B_i(p_i)\}$, where B is p 's bound.

(Necessity) Σ is safe $\Leftrightarrow B(\Sigma_i) = \max_{i=1}^k \{B(p_i)\} = 1 \Leftrightarrow B(p) \leq 1, B(\Sigma) = 1 \Leftrightarrow \Sigma$ is safe.

(Sufficiency) Σ is safe $\Leftrightarrow B(\Sigma) = \max\{B(p)|p \in P\} = 1$

$$\Leftrightarrow \max\{B(p)|p \in P\} = \max\{\min_{i=1}^k B_i(p_i)|p_i \in P_i\} = 1$$

$$\Leftrightarrow p_i \in P_i, B_i(p_i) = 1 \Leftrightarrow \Sigma_i \text{ is safe}$$

To prove the consistency on the fairness, the concepts of repeated vector and repeated Petri net are given.

Definition 18: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, and C is the corresponding incidence matrix of Σ . If there exists n dimensional non-trivial non-negative integer vector such as that of $C^T X \geq 0$. X is a repeated vector of N , where $n = |T|$.

Definition 19: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, if there exists repeated vector in Σ . Then, Σ is a repeated Petri net.

Theorem 5: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, $\Sigma = \{\Sigma_i, T_i; F_i, K_i, W_i, M_{0_i}\} (i = 1, 2, \dots, k)$ is the decomposed subnet. If Σ is a repeated PN, and then each Σ_i is a repeated PN.

Proof: If Σ is a repeated PN. X is one of any repeated vector of Σ . In other words, X is a n dimensional non-trivial non-negative integer vector ($n = |T|$) and $C^T X \geq 0$, where C is an incidence matrix of Σ . Assume C_i is the corresponding incidence matrices of Σ_i . Based on the definition 18, there exists $C = (C_1, C_2, \dots, C_k)$, which satisfies the following equations.

$$C^T X = (C_1, C_2, \dots, C_k)^T X = \begin{pmatrix} C_1^T X \\ C_2^T X \\ \vdots \\ C_k^T X \end{pmatrix} \geq 0 \quad (14)$$

It means that $C_i^T X \geq 0 (i = 1, 2, \dots, k)$. So each Σ_i is repeated Petri net.

Lemma 2[23]: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, C is the corresponding incidence matrix of Σ . The necessary and sufficient condition of Σ is fair is that each repeated vector in Σ does not include zero vector and any two repeated vectors in Σ have a linear correlation.

Theorem 6: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, $\Sigma = \{\Sigma_i, T_i; F_i, K_i, W_i, M_{0_i}\} (i = 1, 2, \dots, k)$, is the subnets. The necessary and sufficient condition of Σ is fair is that each Σ_i is fair.

5. Case Study

In this section, a PN (demonstrated in Figure 3) is used to reveal the feasibility of the proposed decomposition algorithm. Hence, the corresponding index functions of each place are

$$f(p_1) = f(p_4) = 1, f(p_3) = f(p_6) = 2, f(p_2) = f(p_5) = 3$$

Figure 4 shows the corresponding subnets after implementing our algorithm.

a. Analysis on the consistence of dynamic place

Through analysis on the consistence of dynamic place when t_1, t_2, t_3 were enabled, the incidence matrices were calculated as follows.

$$C = \begin{pmatrix} -1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ 1 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} C_1 = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} C_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} C_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

The initial state of each net is described as follows.

$$M_0 = (1 \ 1 \ 1 \ 0 \ 0 \ 0)^{-1} M_{0_1} = (1 \ 0 \ 0 \ 0 \ 0 \ 0)^{-1}$$

$$M_{0_2} = (0 \ 0 \ 1 \ 0 \ 0 \ 0)^{-1} M_{0_3} = (0 \ 1 \ 0 \ 0 \ 0 \ 0)^{-1}$$

After implementing transition sequence $t_1 t_2 t_3 t_4$, then,

$$u = (1 \ 1 \ 1 \ 1)^{-1} u_1 = (1 \ 0 \ 0 \ 1)^{-1} u_2 = (0 \ 0 \ 1 \ 1)^{-1} u_3 = (1 \ 1 \ 1 \ 0)^{-1}$$

Put these conditions into state equations:

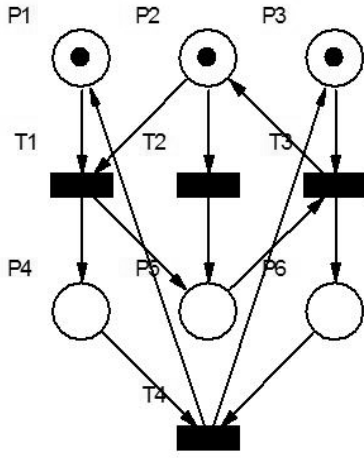


Fig. 3. Original PN model Σ

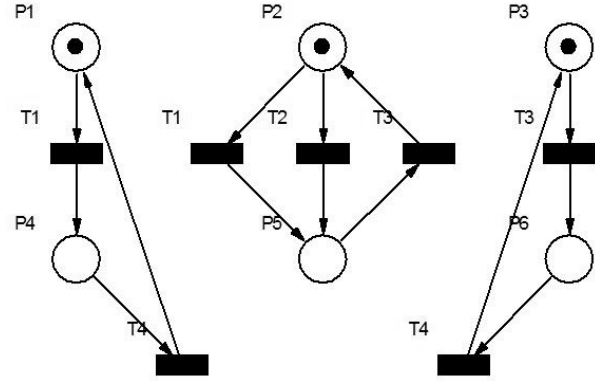


Fig. 4. The decomposed subnets of Fig.3

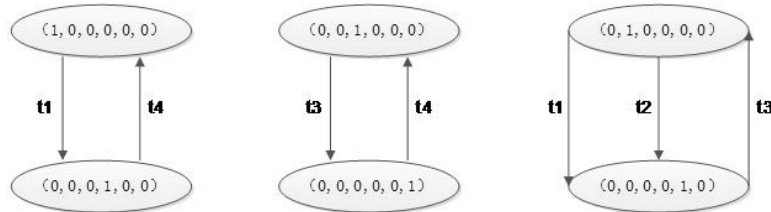


Fig. 5. Each sub-PN's reachable marking graph

$$M = M_0 + Cu \quad M_1 = M_{01} + C_1u_1 \quad M_2 = M_{02} + C_2u_2 \quad M_3 = M_{03} + C_3u_3$$

$$M_0 = (0 \ 0 \ 0 \ 0 \ 1 \ 0)^{-1} \quad M_1 = (0 \ 0 \ 0 \ 0 \ 0 \ 0)^{-1}$$

$$M_2 = (0 \ 0 \ 0 \ 0 \ 0 \ 0)^{-1} \quad M_3 = (0 \ 0 \ 0 \ 0 \ 1 \ 0)^{-1}$$

Then, $M_i(p_i) = M$ ($p_i \in P$) ($i = 1, 2, 3$) can be gained. It means that the composed subnets maintain the unity of dynamic place with original PN model.

b. Analysis on the consistence of liveness and fairness

Lemma 3 [23]: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, if $t \in T$, there exists any $M \in M_0[>$ such that $M'[t >$. Then, t is live.

Lemma 4 [23]: For a PN $\Sigma = \{P, T; F, K, W, M_0\}$, $G(PN)$ is corresponded reachable marking graph. Then,

- (1) If there exist end notes in $G(PN)$, none of one transition is live;
- (2) If transition t in Σ is live, each note in $G(PN)$ will have basic cycle with direct arc marked as t .

Figure 5 shows the reachable marking graphs for each decomposed subnet.

Based on lemma 3 and 4, Σ_1 , Σ_2 and Σ_3 are live.

The conclusions could be obtained that $T_1 \cap T_2 = \{t_4\}$, $T_1 \cap T_3 = \{t_1\}$, $T_2 \cap T_3 = \{t_3\}$. Then, conclusions are reached as follows.

- (1) The reachable marking graphs about Σ_1 and Σ_2 are isomorphism about $\{t_4\}$;
- (2) The reachable marking graphs about Σ_1 and Σ_3 are isomorphism about $\{t_1\}$;
- (3) The reachable marking graphs about Σ_2 and Σ_3 are isomorphism about $\{t_3\}$.

So, Σ is fair, Σ_1 , Σ_2 and Σ_3 are fair respectively.

c. Analysis of the consistency on boundedness and safeness

A PN is safe if there is no more than one token in each place. From figure 3, the conclusion is that \sum_1 , \sum_2 and \sum_3 are all safe nets. So, they are all bounded.

6. Conclusion and future work

This paper proposes a decomposition method to control the scale of PN by using an index function. The framework of this algorithm is given and the implementation process is discussed in an example. This paper also analyzes the consistency on the dynamic properties between the original PN model and corresponding decomposed subnets. The correctness of the proposed algorithm is verified by a case. However, other high-level Petri nets (HLPNs) which are extended based on the PN are also faced with the state explosion issue in the modelling process. So the future work for us is to modify this proposed algorithm which satisfies the characters of other HLPNs.

References

- [1] M. Silva. Half a century after Carl Adam Petri's Ph.D thesis: a perspective on the field [J]. *Ann Rev Control*, 37 (2013) 191–219.
- [2] R. Zurawski, M. C. Zhou. Petri nets and industrial applications: A tutorial [J]. *IEEE Trans.on Industrial Electronics*, 41 (6) (1994) 567-583.
- [3] T. Murata. Petri nets: Properties, analysis, and applications [J]. *Proc.of the IEEE*, 77 (4) (1989) 541-580.
- [4] M. P. Cabasino, A. Giua, C. Seatzu. Diagnosis Using Labeled Petri Nets With Silent or Undistinguishable Fault Events [J]. *IEEE Trans. on Systems, Man and Cybernetics: Systems*, 43 (2) (2013) 345-355.
- [5] F. G. Quintanilla, O. Cardi, A. L'Anton, et al. A Petri net-based methodology to increase flexibility in service-oriented holonic manufacturing systems[J]. *Computers in Industry*, 2016, 76: 53-68.
- [6] F. Jafarnejad, A. A.Pouyan. Distributed Computation of Liveness - Enforcing Supervisory Control of Flexible Manufacturing Systems[J]. *Journal of Advanced Manufacturing Systems*, 2017, 16(01): 1-15.
- [7] W. Liu, L. Wang, Y. Y. Du, & M. Z. Li. Deadlock Property Analysis of Concurrent Programs Based on Petri Net Structure [J]. *International Journal of Parallel Programming*, (2016) 1-20
- [8] L. Wang, Y. Du, & W. Liu. Aligning observed and modelled behaviour based on workflow decomposition [J]. *Enterprise Information Systems*, (2016) 1-21
- [9] W. M. Van der Aalst. Decomposing Petri nets for process mining: A generic approach [J]. *Distributed and Parallel Databases*, 31 (4) (2013) 471-507
- [10] K. Q. Zhou, A. M. Zain, and L. P. Mo. A decomposition algorithm of fuzzy Petri net using an index function and incidence matrix [J]. *Expert Systems with Applications*, 42 (8) (2015) 3980-3990.
- [11] K. Q. Zhou, W. H. Gui, L. P. Mo, and A. M. Zain, A.M. A Bidirectional Diagnosis Algorithm of Fuzzy Petri Net Using Inner-Reasoning-Path. *Symmetry*, 10 (2018), 192.
- [12] Li J, Zhou M C, Dai X. Reduction and refinement by algebraic operations for Petri net transformation. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 42(5) (2012) 1244-1255.
- [13] Shen V R L, Chung Y F, Chen S M, et al. A novel reduction approach for Petri net systems based on matching theory[J]. *Expert Systems with Applications*, 40(11) (2013) 4562-4576.
- [14] Nishi T, Matsumoto I. Petri net decomposition approach to deadlock-free and non-cyclic scheduling of dual-armed cluster tools[J]. *IEEE Transactions on Automation Science and Engineering*, 12(1) (2015)281-294.
- [15] Dideban A, Zeraatkar H. Petri Net controller synthesis based on decomposed manufacturing models. *ISA transactions*, 77 (2018) 90-99.
- [16] Ye J, Zhou M C, Li Z, et al. Structural decomposition and decentralized control of Petri nets[J]. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.
- [17] F. Moutinho, L. Gomes. Asynchronous-channels within Petri net-based GALS distributed embedded systems modeling [J]. *IEEE Transactions on Industrial Informatics*, (2014) 10 (4) 2024-2033
- [18] T. Nishi, I. Matsumoto. Petri net decomposition approach to deadlock-free and non-cyclic scheduling of dual-armed cluster tools [J]. *IEEE Transactions on Automation Science and Engineering*, (2015) 12(1) 281-294
- [19] T. Nishi, Y. Tanaka. Petri net decomposition approach for dispatching and conflict-free routing of bidirectional automated guided vehicle systems [J]. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 42 (5) (2012) 1230-1243
- [20] M. Vatani, A. Doustmohammad. Decomposition of first-order hybrid Petri nets for hierarchical control of manufacturing systems [J]. *Journal of Manufacturing Systems*, 35 (2015) 206-214
- [21] K. Q. Zhou, A. M. Zain, and L. P. Mo. Dynamic properties of fuzzy Petri net model and related analysis [J]. *Journal of Central South University*, 22 (12) (2015) 4717-4723
- [22] K. Q. Zhou, A. M. Zain, Fuzzy Petri nets and industrial applications: a review[J]. *Artificial Intelligence Review*, 2016, 45(4): 405-446.
- [23] Z. H. Wu, Introduction to Petri Net [M]. China Machine Press (2006).