TECHNICAL RESEARCH REPORT

A Class of Conflict-Free Petri Nets Used for Controlling Manufacturing Systems

by G. Harhalakis, M. Leventopoulos, C-P. Lin, R. Nagi, and J-M. Proth

T.R. 92-90r2

INSTITUTE FOR SYSTEMS RESEARCH

Sponsored by
the National Science Foundation
Engineering Research Center Program,
the University of Maryland,
Harvard University,
and Industry

A Class of Conflict-Free Petri Nets used for Controlling Manufacturing Systems†

G.Harhalakis[1], M.Leventopoulos[2], C.-P.Lin[1], R.Nagi[3], J.-M.Proth[4]

[1]Department of Mechanical Engineering and Institute for Systems Research, University of Maryland, College Park, MD 20742, U.S.A.
Internet: george/lincp@src.umd.edu

[2]Department of Computer Science and Institute for Systems Research, University of Maryland, College Park, MD 20742, U.S.A.
Internet: mml@cs.umd.edu, Student member IEEE.

[3]Department of Industrial Engineering, SUNY at Buffalo, 342 Bell Hall, Buffalo, NY 14260, U.S.A.
Internet: nagi@eng.buffalo.edu

[4]INRIA, 4 rue Marconi, Technopole Metz 2000, 57070 Metz, FRANCE, and Institute for Systems Research, University of Maryland, U.S.A.
Internet: proth@ilm.loria.fr, Member IEEE.

Abstract

This paper is devoted to the behavior, evaluation, and management of non-cyclic discrete systems in general and manufacturing systems in particular. We introduce a special type of Petri nets called CFIOs (Conflict-Free nets with Input and Output transitions). It is shown that CFIOs are live, reversible if consistent, and can be kept bounded under certain conditions. We also develop reduction rules which facilitate the computation of the t-invariants of CFIOs. We then take advantage of the qualitative properties of CFIOs to perform planning in manufacturing systems. Numerical examples illustrate these approaches.

Key words: Petri nets, Discrete event systems, Manufacturing systems, Consistency, Boundedness, Reduction of Petri nets, Conflict-free systems, t-invariants, Planning

1 Introduction

Significant changes have been occurring in industrialized countries during the past three decades. To name only a few of them: (i) production is moving towards high quality products and small lot sizes (it is said that the economy of scale has been replaced by the economy of scope), (ii) competitiveness has become a worldwide phenomenon, and (iii) erratic demands are progressively replacing steady demands.

One way to approach these issues is to introduce highly automated manufacturing systems with high flexibility and small, or even zero, set-up times. Such systems are highly sophisticated.

† Supported in part by the NSF under Grant CDR-88-03012

They require expensive components, are costly to run and maintain and, at the beginning of their life cycle, demand a very complex design process.

The first stage of this process is the so-called preliminary design [2], which includes the tasks to be performed in order to specify the manufacturing system given the parts' specifications. This stage is also known as the *paper study,* and ends when the physical implementation of the system starts. Among the tasks to be performed at this level are the establishment of the functional specifications, the modeling and the evaluation of the manufacturing system. Many tools exist to support these tasks. For instance, queueing theory, state-transition, mathematical programming and simulation, are used for the performance evaluation of the system, while entity-relationship approaches and the CIMOSA[1] related tools are used for the specification of the system.

It is acknowledged that Petri nets are one of the few tools which can support specification as well as functional modeling and performance evaluation (see also [7]). The latter is performed by using a Petri net based simulation software or by using the properties of Petri nets, which are particularly rich in some applications. Thus Petri nets can be considered as a tool that enables a step towards integration in the preliminary design stage.

Numerous analytical results (see in particular [1], [3], [4], [5], [8] and [9]) are available to support the preliminary design of cyclic manufacturing systems. The reason is that these kinds of manufacturing systems can be modeled using strongly connected event graphs, a special class of Petri nets whose analytical properties are particularly powerful.

However, strongly connected event graphs cannot be employed for non-cyclic manufacturing systems. The main reasons for this are: (i) we have to be able to introduce and remove parts from the system (and this process cannot be represented by a strongly connected net), and (ii) the system is not *conflict-free* (due to the decision making process).

Non-cyclic manufacturing systems are modeled using general Petri nets enhanced by:

i. **Input transitions**, which can deliver tokens to the system, making it possible to model raw material and semi-finished products originating outside the manufacturing system.
ii. **Output transitions** which can remove from the system tokens that represent finished or semi-finished products leaving the system.

As soon as a control decision is applied, some of the transitions are *frozen* (i.e. cannot be fired anymore). Appropriate control turns the Petri net model into a conflict-free net. This conflict-free net is obtained by removing from the initial Petri net model the frozen transitions, the related arcs, and their input and output places, if they are not connected to a non-frozen transition. Such a conflict-free subnet of the initial Petri net model is referred to hereafter as a CFIO net, which stands for Conflict-Free with Input and Output transitions. A sequence of decisions applied to a non-cyclic manufacturing system can thus be viewed as a sequential activation of conflict-free subnets of the initial Petri net model.

In this paper, we focus on the study of CFIOs. Such a subnet is characterized by input and output transitions and by the fact that each place has only one output transition. We will show that consistency is a key property for CFIOs, as modeling realistic manufacturing systems with Petri nets decomposable to consistent CFIOs guarantees the desired qualitative aspects of the system.

[1] Computer Integrated Manufacturing — Open System Architecture, ESPRIT program

An algorithm that tests a CFIO for consistency and also computes the underlying *t-invariants* of the CFIO will be presented. We will further show that those models, along with the t-invariant information, can be used for computing the planning of the related manufacturing system.

In order to facilitate control of the manufacturing system, its net should be decomposable into manageable CFIOs. By *manageability* we mean the following:

i. We are in a position to keep the CFIO net bounded, which means that the work-in-process of the related manufacturing system can be held below a given level.
ii. Any marking reachable from the initial marking can be reached from any other marking reachable from the initial marking; this means that the related manufacturing system can always reach any of the states for which it was designed.
iii. The ratios of the output transition firings can vary to a large extent, which guarantees the flexibility of the manufacturing system at hand.

The remainder of this paper is organized as follows: In Section 2 we provide the definitions used in the paper. In Section 3, we look at manageability from a Petri net point of view. We analyze the qualitative properties of CFIO nets and relate them to manageability as defined above. In Section 4, we introduce a process to transform a CFIO N into a reduced CFIO N_1 such that N is consistent if and only if N_1 is consistent, and the t-invariants of N can be derived from the t-invariants of N_1. We also propose an algorithm that derives the reduced net, uses it to compute the t-invariants of the initial net, and decides if the initial net is consistent or not. An example is presented to illustrate this algorithm. In Section 5, we use CFIOs to perform planning in a manufacturing system. The approach is illustrated by a small example. Section 6 presents our conclusions.

2 Definitions

In this section, we define generic Petri nets ([6],[14]) and provide the definition of the CFIO nets proposed in this paper.

2.1 Definitions related to Petri Nets

A Petri net can be viewed from two aspects: static and dynamic. The static aspect of a Petri net is defined by a weighted, bipartite, directed graph which consists of places (represented by circles), transitions (represented by bars), and arcs (represented by arrows). The dynamic aspect is provided by the initial and subsequent markings of the net, as enabled transitions are fired.

Definition 1: A Petri-net is a four-tuple $N =< P, T, F, W >$ where :

- $P = \{p_1, p_2, \ldots, p_n\}$ is a finite set of places,
- $T = \{t_1, t_2, \ldots, t_m\}$ is a finite set of transitions,
- $P \cap T = \emptyset$; i.e. places and transitions are disjoint sets,
- $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs (flow relations),
- $W : F \rightarrow \mathbf{N}^+$ is a weight function which assigns a strictly positive value to each arc.

In the case that all weights are equal to one, the net is called *ordinary*.

A place $p \in P$ is an input (resp. output) place of $t \in T$ if there exists a directed arc joining p to t (resp. t to p). The set of input places of t is denoted by ${}^\bullet t$, while the set of output places of t is denoted by $t^\bullet$. Similarly, $t \in T$ is an input (resp. output) transition of $p \in P$ if there exists a directed arc joining t to p (resp. p to t). The set of input (resp. output) transitions of p is denoted by ${}^\bullet p$ (resp. $p^\bullet$).

This notation can be extended to a subset of places or transitions. For example ${}^\bullet S$ with $S \subseteq P$, is the union of all ${}^\bullet p$ such that $p \in S$. Accordingly ${}^\bullet T$ (resp. $T^\bullet$) denotes the set of all the places of the net which are input (resp. output) places of at least one transition. ${}^\bullet P$ (resp. $P^\bullet$) denotes the set of all the transitions which are input (resp. output) transitions of at least one place.

Definition 2: A *marked Petri net* is a Petri net $PN =< N, M >$ where:

- $N =< P, T, F, W >$ is a Petri net as defined above,
- $M : P \rightarrow \mathbb{N}$ is called a marking of PN.

For $p \in P, M(p)$ is the number of tokens contained in the place p for marking M. In Figure 1, we present an ordinary Petri net where tokens are represented by dots.

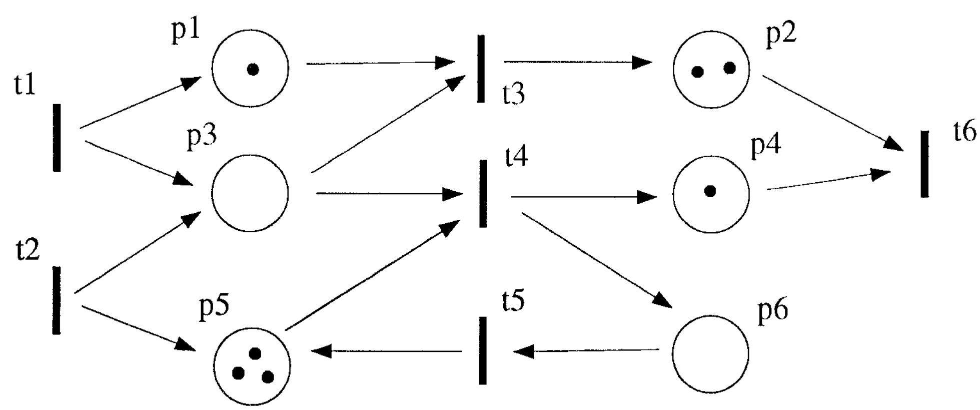

Figure 1: A marked Petri net

The marking represented in Figure 1 is $M =< 1, 2, 0, 1, 3, 0 >$, which means that $M(p_1) = 1, M(p_2) = 2$ and so on.

Definition 3: A *pure* Petri net $N =< P, T, F, W >$ is such that ${}^\bullet t \cap t^\bullet = \emptyset$, $\forall t \in T$.

A pure and ordinary Petri net can be represented by an incidence matrix $C = [c_{ij}], i = 1, 2, \ldots, n; j = 1, 2, \ldots, m,$ where

$$c_{ij} = \begin{cases} -1, & \text{if there is an arc from place i to transition j} \\ +1, & \text{if there is an arc from transition j to place i} \\ 0, & \text{otherwise} \end{cases}$$

As an example, the following incidence matrix corresponds to the Petri net presented in Figure 1.

$$C = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 1 & 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \end{bmatrix}$$

Note: For the remainder of this paper, unless otherwise noted, we consider only *pure* and *ordinary* Petri nets.

In a marked Petri net, a transition $t \in T$ is said to be *enabled* if every $p \in {}^\bullet t$ contains at least one token. *Firing* an enabled transition consists of removing one token from each of the input places and adding one token to each output place (i.e. to every place $p \in t^\bullet$).

For instance, in the Petri net given in Figure 1, t_6 is enabled. Firing t_6 consists only of removing one token from p_2 and p_4; because $t_6^\bullet = \emptyset$, no place receives a token.

We can also associate a *firing time* to each transition. Such a time represents the time between the beginning and the end of a transition firing. Since transitions model operations in a manufacturing system, firing times represent manufacturing times. Tokens are reserved in the input places during the firing process. They are removed from the input places and the output places are marked at the end of the firing process.

Definition 4: *Siphon* and *Trap:* In an ordinary net, a non-empty subset of places S is called a *siphon* if ${}^\bullet S \subseteq S^\bullet$. It is called a *trap* if $S^\bullet \subseteq {}^\bullet S$.

By firing a sequence of transitions σ, we can reach a marking M from an initial marking M_0. We say that M is *reachable* from M_0, and that σ is a *firing sequence* starting at M_0 and leading to M. We denote this as follows:

$$M \in R(M_0)$$

and

$$M_0 \xrightarrow{\sigma} M$$

A basic relationship in Petri net theory is that:

$$M = M_0 + C\alpha_\sigma \tag{1}$$

where $\alpha_\sigma = [\alpha_\sigma^1, \alpha_\sigma^2, \ldots, \alpha_\sigma^m]$ and α_σ^i is the number of times transition t_i is fired in σ. α_σ is called the *counting vector*.

An *elementary circuit* is a directed path which starts from a node (place or transition) of the Petri net, ends at the same node, and is such that no node of the path is included more than once. For instance, <p_5, t_4, p_6, t_5, p_5> is an elementary circuit of the net of Figure 1.

A Petri net marking M reflects the state of the manufacturing system modeled by the net. Hereafter, the terms state of the system and Petri net marking will be interchangeable.

In manufacturing systems, tokens can be used for representing parts, resources, or other information, while places represent buffers, and transitions represent operations.

2.2 Definition of a CFIO Net

A *CFIO net* is a subclass of ordinary and pure Petri nets with the following three properties:

1. It is structurally *conflict-free:*

$$^\bullet t_1 \cap {}^\bullet t_2 = \emptyset, \; \forall t_1, t_2 \epsilon T$$

In other words, each place has only one output transition.

2. It has neither input nor output places (also called source/sink places):

$$^\bullet T = T^\bullet$$

3. It has at least one input and at least one output transition:

$$T - P^\bullet \neq \emptyset$$
$$T - {}^\bullet P \neq \emptyset$$

The net represented in Figure 2 is a CFIO net, where t_1 and t_2 are the input transitions, and t_8 and t_9 are the output transitions.

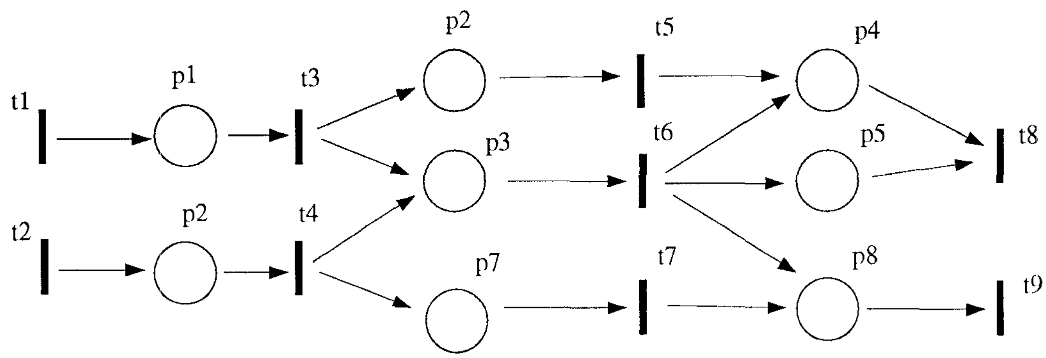

Figure 2: A CFIO net

3 Properties of CFIO nets

In the following section we consider liveness, boundedness, and consistency of CFIOs, and we will show their relationship to manageability.

Note that a CFIO net is structurally conflict-free. From the definition of CFIOs it follows that they are structurally live. We show that CFIOs are not structurally bounded but can be kept bounded if the underlying model is a realistic manufacturing system. We will see that reversibility is a consequence of consistency in this particular case, and that the net is reversible for any initial marking where every elementary circuit has at least one token.

3.1 Liveness

A Petri net is said to be *structurally live* if there exists at least one marking M_0 for which the marked net is live, i.e. for all $t \in T$ and for all $M_1 \in R(M_0)$, there exists a marking $M_2 \in R(M_1)$ such that t is enabled in M_2.

Result 1: A CFIO net is structurally live.

Proof: For this proof the following known theorem is used[2]:

Theorem: A conflict-free net N is structurally live *iff* every siphon in N contains a trap.

Note that every circuit in a conflict-free net is a trap. Thus it suffices to show that every siphon in N contains a circuit.

Let $N =< P, T, F, W >$ be a CFIO net. Consider any subset S of P such that S forms a *siphon* (if N contains no siphons we are done) and let N_S be the subnet of N consisting of the places in S along with the connected transitions and arcs.

Note that N_S is also a CFIO net, thus it has no input places. Moreover N_S has no input transitions, since its places form a siphon. Since N_S has no input at all, it must contain a circuit.

Q.E.D.

Corollary 1: Every siphon in a CFIO net contains a circuit.

Note: If N is a structurally live Petri net, then there exists a vector x of positive integers such that $Cx \geq 0$ (for details on structural properties see [12], [13]).

3.2 Boundedness

A Petri net is said to be *structurally bounded*, if for any initial marking M_0, there exists an integer $k > 0$ such that for any $M \in R(M_0), M(p) \leq k, \forall p \in P$.

Result 2: A CFIO net is not structurally bounded.

Proof: A CFIO net has at least one input transition, thus at least one column of the incidence matrix contains only non-negative values, and at least one of these values is 1. If $[a]$ is this column, then $\nexists y > 0$ such that $y^T[a] \leq 0$. Thus, $\nexists y > 0$ such that $y^T C \leq 0$.

Q.E.D.

Note: A Petri net is structurally bounded if and only if there exists a vector y of positive integers such that $y^T C \leq 0$.

Result 3: A CFIO net can be kept bounded by controlling the firings of the transitions belonging to $T - P^\bullet$ (i.e. the input transitions) if and only if the Petri net obtained by removing those transitions is bounded.

Proof:

(a) The condition is sufficient: Let $A = [A_1 A_2]$ be the incidence matrix of the CFIO net where A_1 is composed with the columns of $T - P^\bullet$. Thus the columns of A_1 do not contain (-1) values.

2 In Murata [6] this is shown for free-choice nets, a superset of conflict-free nets.

Boundedness of the net defined by A_2 means that :

$$\exists K_1 = [k_1, k_1, \ldots, k_1]^T \ , \ \ k_1 \in \mathbf{N}^+ \text{ s.t. } M_0 + A_2\alpha_{\sigma_2} \leq K_1$$

for any initial marking M_0 and for any feasible sequence σ_2 of transition firings related to $P^\bullet$, α_{σ_2} being the counting vector.

But, for any $K = [k, k, \ldots, k]^T \ , \ \ k > k_1$, there exists a feasible sequence σ_1 of transition firings related to transitions $T - P^\bullet$ such that:

$$A_1\alpha_{\sigma_1} \leq K - K1$$

Thus, $\exists K_1$ and σ_1 such that:

$$M_0 + [A_1 A_2]\begin{bmatrix}\alpha_{\sigma_1} \\ \alpha_{\sigma_2}\end{bmatrix} \leq K$$

and the CFIO net can be kept bounded for a given σ_1 for any initial marking M_0.

(b) The condition is necessary: If A_2 as defined above, is not bounded, then there does not exist $K_1 > 0$ such that:

$$M_0 + A_2\alpha_{\sigma_2} \leq K_1$$

for at least one feasible sequence σ_2 related to an initial marking M_0.

Also, because $A_1\alpha_{\sigma_1} \geq 0$, there does not exist $K > 0$ such that:

$$M_0 + A_1\alpha_{\sigma_1} + A_2\alpha_{\sigma_2} = M_0 + A\alpha_\sigma < K$$

and thus the CFIO net cannot be kept bounded.

Q.E.D.

Note: From Result 3 it can be deduced that if the net obtained by removing the input transitions remains unbounded, the initial net is also unbounded and thus non-manageable.

Consider such a net N and let N_2 be the net obtained from N by removing the input transitions. N_2 is unbounded, at some place p, implying that one of the inputs of p, say t, can fire an infinite number of times (for some initial marking).

Since there are no input transitions, t must drain a token (an infinite number of times) from some other place, and since the initial (and every other) marking is finite, t must belong to a circuit and "feed" its input place (indirectly) by its own firings.

This implies that t belongs to a marked circuit and, furthermore, to a circuit in which all of its transitions can be enabled an infinite number of times. This further implies that all of the input places of the transitions of that circuit belong to a circuit; otherwise they would not be able to supply an infinite number of tokens.

We have thus shown that if N_2 is unbounded, it must contain a circuit (not necessarily elementary) such that all of its transitions have inputs only within the circuit. We will call such a circuit *self-sustained.*

In the manufacturing system such a situation is highly unlikely as it implies the presence of an uncontrollable subsystem. Therefore a CFIO remaining unbounded after we remove (or effectively control) its input transitions probably indicates some design or modeling error. We will see in the next section how to detect such problems that lead to an unmanageable model.

3.3 Consistency

A CFIO net is *consistent* if there exists a marking M_0 and a transition firing sequence σ where every transition occurs at least once in σ and $M_0 \xrightarrow{\sigma} M_0$.

We know ([6],[14]) that consistency holds if and only if there exists a vector x of positive integers such that $Cx = 0$.

Result 4: If a CFIO net is consistent, then for any initial marking M_0 such that each elementary circuit contains at least one token, and for every $M \in R(M_0)$, we have $M_0 \in R(M)$ (i.e. the net is reversible for every initial marking M_0 such that all elementary circuits are marked).

Proof: In this proof we will use the following known theorem[3]:

Theorem: In a conflict-free net N, M_d is reachable from M_0 if there exists a non-negative integer vector $x =< x_1, x_2, \cdots, x_m >$ such that $M_d = M_0 + Cx$, and every siphon in (N_x, M_{ax}) contains a marked trap. C denotes the incidence matrix of N. N_x denotes the subnet of N consisting of transitions t such that $x_t > 0$ along with their input and output places and their connecting arcs. M_{ax} denotes the subvector of M_a for places in N_x.

By Corollary 1, and since all circuits are marked, every siphon in N contains a marked trap. M is reachable from M_0, therefore there exists a non-negative integer vector x such that:

$$M = M_0 + Cx$$

By assumption N is consistent thus there exists a positive vector z such that $Cz = 0$.

Without loss of generality we can select z large enough so that $z > x$ (by multiplying it by an arbitrarily large constant). Let $y = z - x$. We then have $Cx = -Cy$ and $M_0 = M + Cy$.

It remains to show that every siphon in N_y contains a marked trap under the marking M. Note however that $N_y = N$ because $y > 0$, and all siphons in N contain a marked trap in all markings reachable by M_0. Thus M_0 is reachable from M.

Q.E.D.

Corollary 2: For every initial marking M_0 such that all elementary circuits contain at least one token each, $M_1 \in R(M_0)$ and $M_2 \in R(M_0)$, then $M_1 \in R(M_2)$ and $M_2 \in R(M_1)$.

This corollary means that, in a consistent CFIO net, for any initial marking M_0 with at least one token in every elementary circuit, any marking reachable from M_0 can be reached from any other marking reachable from M_0. Result 4 shows that the consistency of a CFIO net implies item *(ii)* of manageability as described in the introduction.

[3] Theorem 18 of Murata [6], which considers TCC nets, a superclass of conflict-free nets.

4 Reduction of CFIO nets

From the previous section we have seen that in order for a CFIO net to be manageable

(i) all of its elementary circuits have to be marked in the initial marking,

(ii) it should not contain any self-sustained circuits,

(iii) it should be consistent.

The first requirement is directly controllable by the manufacturer. A simple test will be described for the second requirement after we introduce the circuit reduction technique.

Consistency is a key property of all manageable CFIO nets. Thus, we should be able to verify this property even for large-scale nets in a reasonable amount of time. We also need to compute all the minimal t-invariants of the CFIO net. This allows control of the CFIO net according to external criteria. In this subsection, we propose two results which show that it is always possible to transform a CFIO net N into a reduced CFIO net N^1 which is consistent if and only if N is consistent. N^1 has only input and output transitions which correspond to the input and output transitions of N.

4.1 Circuit Reduction

Result 5: Let N be a CFIO net.

Case 1: If, for at least one elementary circuit $\gamma \subseteq N$, there exists a place $p \in \gamma$ which has more than one input transition, then N is not consistent.

Case 2: Otherwise, if each elementary circuit is marked, then the net N^1 is consistent if and only if N is consistent. N^1 is derived from N by replacing each elementary circuit γ by a transition t_γ such that any input (resp. output) place of a transition of γ which is not included in γ becomes an input (resp. output) place of t_γ.

Proof of Case 1: Assume that, for an elementary circuit $\gamma \subseteq N$, there exists a place $p \in \gamma$ which has more than one input transition. Assume also that N is consistent.

According to the definition of consistency, there exists a marking M_0 and a transition firing sequence σ where every transition occurs at least once in σ, such that $M_0 \xrightarrow{\sigma} M_0$. Since N is conflict-free, $|p^\bullet| = 1$ and $p \in \gamma \Rightarrow p^\bullet \in \gamma$. This implies that the total number of tokens in the circuit γ cannot be decreased. Now if there exists $p \in \gamma$ such that $|{}^\bullet p| > 1$, firing once every transition $t \in {}^\bullet p$ will increase by $|{}^\bullet p| - 1$ the number of tokens in γ. Since σ contains each $t \in {}^\bullet p$ at least once, firing σ will increase the number of tokens in γ. Thus M_0 cannot be recovered and N is not consistent.

In Figure 3, we give two examples of such inconsistent CFIO nets.

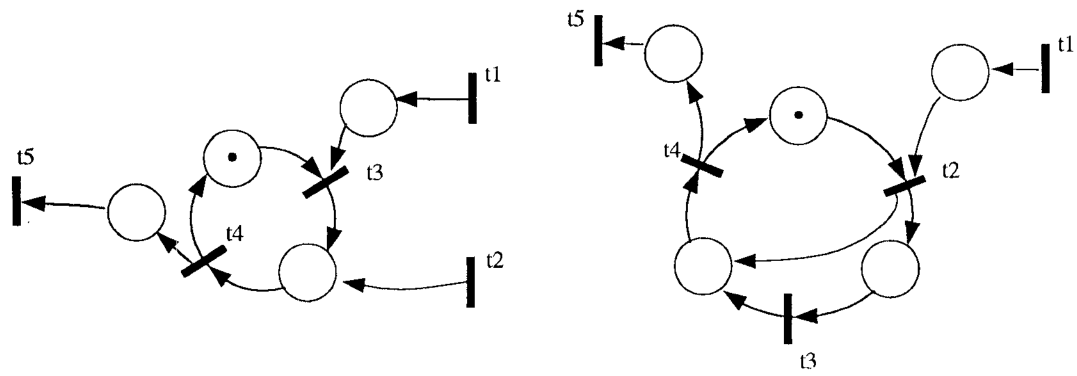

Figure 3: Two inconsistent CFIO nets

Proof of Case 2: Let us now assume that each elementary circuit contains at least one token and that, in any elementary circuit, a place has only one input transition. We want to prove the reduction rule presented in Figure 4.

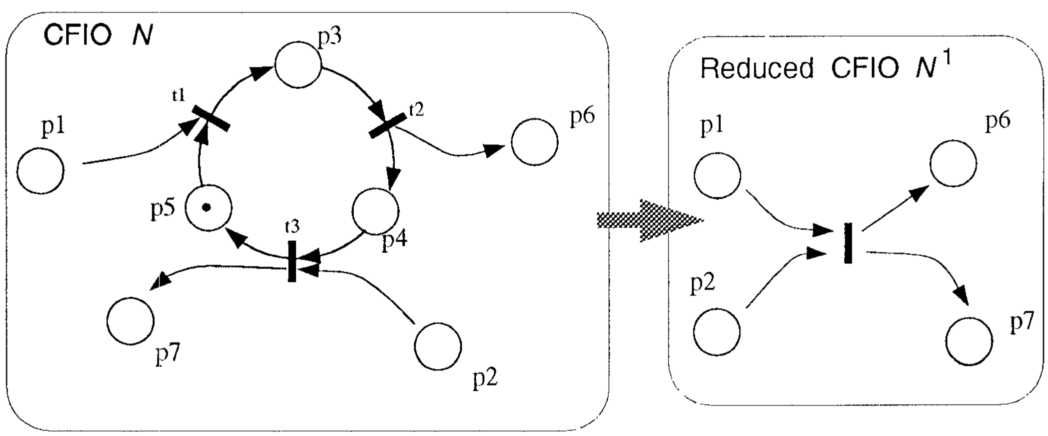

Figure 4: First reduction rule

In the remainder of the proof, C and C^1 are the incidence matrices of N and N^1 respectively. The proof consists of two parts:

(a) If N is consistent then N^1 is consistent.
(b) If N is not consistent then N^1 is not consistent.

(a) *N is consistent:*

According to the definition of the consistency, $\exists x \in \mathbb{N}^m, x > 0$ such that $Cx = 0$. Without loss of generality, we can assume that N contains only one elementary circuit $< p_{i_1}, t_{j_1}, p_{i_2}, t_{j_2}, ..., p_{i_s}, t_{j_s} >$. We assume that the rows (resp. the columns) of C are ordered according to the indices of the places (resp. the transitions).

C^1 is derived from C as follows:

Derive C^a from C by setting:

$$c^a_{i,j} = \begin{cases} c(i,j) & \text{for j} \neq j_1 \\ \sum_{k=1}^{s} c(i,j_k) & \text{for } j = j_1 \end{cases}$$

Derive C^b from C^a by removing the columns of transitions $j_2, j_3, ..., j_s$.
Obtain C^1 by removing the rows of places $i_1, i_2, ..., i_s$ of C^b.
Furthermore, each row $i \in \{i_1, ... i_s\}$ in C is such that:

$$c_{i,j} = \begin{cases} -1, & \text{for exactly one } j \in \{j_1, ..., j_s\}, \text{ say } j_i^- \\ +1, & \text{for exactly one } j \in \{j_1, ..., j_s\}, \text{ say } j_i^+ \\ 0, & \text{otherwise} \end{cases}$$

and $Cx = 0$ implies:

$$\begin{aligned} &\sum_{j=1}^{m} c_{i,j} x_j = 0 \text{ for } i = 1, 2, \ldots, n \\ \Longrightarrow\, & c_{i,j_i^-} x_{j_i^-} + c_{i,j_i^+} x_{j_i^+} = 0 \text{ for } i \in \{i_1, i_2, \ldots, i_s\} \\ \Longrightarrow\, & x_{j_i^-} = x_{j_i^+} \end{aligned}$$

This being true for any $i \in \{i_1, ... i_s\}$, it leads to: $x_{j_1} = x_{j_2} = \cdots = x_{j_s}$
Let us now build x^1 as follows:

(i) if $j \notin \{j_1, j_2, ... j_s\}$,
$x^1_k = x_j$, if the k^{th} column of C^1 has been derived from the j^{th} column of C.
(ii) $x^1_v = x_{j_1}$, where v is the rank in C^1 of the column obtained by adding the columns $j_1, j_2, ..., j_s$ of C.

Based on the process followed to build C^1 and the definition of x^1, $Cx = 0$, we obtain $C^1 x^1 = 0$, and N^1 is consistent.

Q.E.D.

(b) *N is not consistent:*

If N^1 were consistent, then:

$$\exists x^1 \in \mathbf{N}^{m_1}, x^1 > 0 \text{ s.t. } C^1 x^1 = 0$$

where m_1 is the number of transitions of N^1.
It would then be possible to expand x^1 to obtain $x \in \mathbf{N}^m$ by setting:

i. $x_j = x^1_k$
if the k-th column in C^1 is the j-th column in C.
ii. $x_{j_1} = x_{j_2} = \ldots = x_{j_s} = x^1_k$
if transition t_k replaces the elementary circuit $\gamma \subseteq N$ containing $t_{j_1}, t_{j_2}, ..., t_{j_s}$.

Using the same arguments as in **(a)**, we can see that $x > 0$ and $Cx = 0$.

Q.E.D

Note: *Detection of Self-Sustained circuits*: As mentioned in the previous section, all input places of a self-sustained circuit are from within the circuit itself. Circuit reduction, as described above, will reduce such a circuit to an input transition, facilitating thus the easy detection of self-sustained circuits in the model.

4.2 Path algorithm

From now on, we assume that the CFIO net at hand does not contain any elementary circuits.

For each transition t, we denote by n_t the number of times t has been fired. Assuming that initially there is no token in the CFIO net (i.e. the initial marking is zero), the following inequalities hold:

$$n_t \leq \sum_{u \in {}^{\bullet}p} n_u, \quad \forall p \in {}^{\bullet}t \tag{2}$$

If we apply relation (2) repetitively, starting from an output transition t_j, we obtain a set of inequalities where only the numbers of input and output transition firings are involved:

$$n_{t_j} \leq \sum_{k \in E} \beta_k^s n_{t_k} \qquad \text{for } s = 1, 2, \ldots, H_j \text{ and } j \in \mathrm{F} \tag{3}$$

where:

- E = $\{k | t_k$ is an input transition$\}$
- F = $\{j | t_j$ is an output transition$\}$
- H_j refers to the number of different inequalities obtained starting from output transition t_j.
- β_k^s are positive integers.

Inequalities (3) provide the maximal number of times an output transition can fire, knowing the number of times the input transitions have been fired, assuming that the initial marking is zero. Furthermore:

(i) The CFIO net is consistent only if it is possible to remove from the CFIO net the tokens which have been introduced by firing the input transitions, assuming that each input transition has been fired at least once.

According to the process which leads to inequalities (3), a necessary and sufficient condition for the CFIO net to be consistent is then to find $n_{t_k} > 0$ for every $k \in E$ and $n_{t_j} > 0$ for every $j \in F$ such that inequalities (3) turn into equalities.

Similarly, the CFIO net has a t-invariant if there exists $n = [n_{k_1}, n_{k_2}, \ldots, n_{k_L}], n \ngtr 0$, $\{k_1, k_2, \ldots, k_L\} = E \cup F$, such that the inequalities (3) turn into equalities. Note that if a Petri net is consistent, it has at least one t-invariant, but the converse is not true.

(ii) It is possible, starting from inequalities (3), to construct a reduced CFIO net with the following properties:

- input transitions are those of the initial net,

- output transitions are also those of the initial net,
- it does not contain any other transition other than the input and output transitions

The incidence matrix C^1 of this reduced net is obtained starting from the inequalities derived from (3), i.e.:

$$-n_{t_j} + \sum_{k \in E} \beta_k^s n_{t_k} = 0 \qquad (3-\text{bis})$$

Each inequality provides one row of C^1.
In the row related to relation (3–bis) (i.e. the s^{th} relation related to t_j) :

- the element of C^1 corresponding to output transition t_j is (-1),
- for every $k \in E$, the element of C^1 corresponding to input transition t_k is β_k^s,
- the other elements of the row are 0.

This reduced CFIO net is hereafter denoted by N^1, while the initial CFIO net is denoted by N.

Like N, N^1 is consistent if inequalities (3) turn into equalities for strictly positive values. N^1 has a t-invariant if, like N, inequalities (3) turn into equalities for values which are positive or zero, with at least one of them being positive.

Assume that we know x^1 ($x^1 > 0$ or $x^1 \not\geq 0$) such that $C^1 x^1 = 0$. Firing the input transitions of N as many times as indicated in x^1 for the corresponding input transition of N^1, and then firing the enabled transitions until none remains enabled anymore, we produce the following results :

(i) the vector x whose components represent the number of times each transition fired is such that $Cx = 0$

(ii) the components of x and x^1 corresponding to the input and output transitions are the same.

The following result summarizes the above remarks:

Result 6: Given a CFIO net N, it is always possible, by repeatedly applying inequalities (2), to upper bound the number of times each output transition can fire by a set of positive linear combinations of the number of times each input transition has been fired, assuming that the initial marking of N is a zero-vector (see inequalities (3)). From these upper bounds, it is possible to derive a reduced net N^1 as defined previously. Based on the fact that N and N^1 are CFIO nets, N is consistent if and only if N^1 is consistent.

Furthermore:

(i) A vector $x > 0$ such that $Cx = 0$ is derived in an unique way from a vector $x^1 > 0$ such that $C^1 x^1 = 0$, and vice versa, where C (resp. C^1) is the incidence matrix of N (resp. N^1).

(ii) A vector $x \not\geq 0$ such that $Cx = 0$ is derived in a unique way from a vector $x^1 \not\geq 0$ such that $C^1 x^1 = 0$, and vice versa. Vector x (resp. x^1) is a t-invariant of N (resp. N^1).

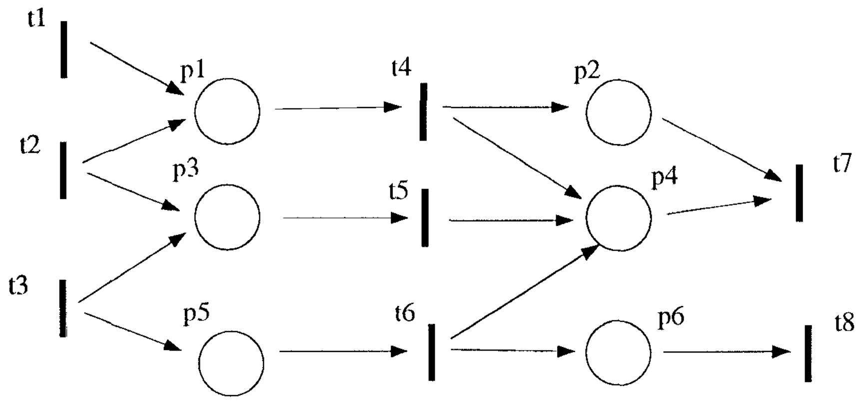

Figure 5: Initial net N

Example: Consider the CFIO net given in Figure 5. First we apply relation (2) to the output transition t_7 and we obtain:

$$n_7 \leq n_4 \qquad (4-1)$$

$$n_7 \leq n_4 + n_5 + n_6 \qquad (4-2)$$

Considering t_4, t_5 and t_6, we obtain, respectively:

$$n_4 \leq n_1 + n_2 \qquad (4-3)$$

$$n_5 \leq n_2 + n_3 \qquad (4-4)$$

$$n_6 \leq n_3 \qquad (4-5)$$

Combining (4–3) with (4–1), and (4–3) through (4–5) with (4–2), we obtain, respectively:

$$n_7 \leq n_1 + n_2 \qquad (5-1)$$

$$n_7 \leq n_1 + 2n_2 + 2n_3 \qquad (5-2)$$

Similarly, starting from t_8, we obtain:

$$n_8 \leq n_3 \qquad (5-3)$$

Inequalities (5-1) to (5-3) comply with the inequalities (3).

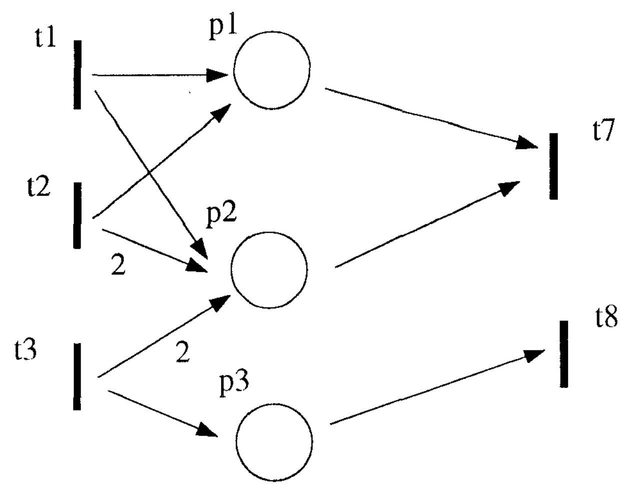

Figure 6: Reduced net N^1

The related reduced net N^1 is given in Figure 6.

In this example:

$$C = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{array} \begin{array}{c} \begin{array}{cccccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \end{array}$$

and

$$C^1 = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \end{array} \begin{array}{c} \begin{array}{ccccc} t_1 & t_2 & t_3 & t_7 & t_8 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & -1 & 0 \\ 1 & 2 & 2 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix} \end{array}$$

N^1 is not consistent, because there is no solution of $C^1x^1 = 0$ such that $x^1 > 0$. Thus, N is not consistent. However, vector $x^1 = [1, 0, 0, 1, 0]$ is such that $C^1x^1 = 0 : x^1$ is a t-invariant, which means that if we fire t_1 once starting from a zero-marking, then we come back to a zero marking by firing t_7.

We see that:

$n_1 = n_7 = 1$ and $n_2 = n_3 = n_8 = 0$ is a solution which turns inequalities (4) into equalities.

Finally we can see that, by firing t_1 in N, we can fire t_4 and t_7. Thus $x = [1, 0, 0, 1, 0, 0, 1, 0]$ is a t-invariant of N.

Note:

(i) the number of places in the reduced net is equal to the number of inequalities (3),

(ii) The reduced net is *not necessarily* an ordinary net; for instance, the weights of (t_2, p_2) and (t_3, p_2) equal to 2 in this example.

4.3 Computation of the minimal t-invariants

The set of transitions corresponding to nonzero entries in a t-invariant is called a *support* of the invariant. A support is said to be *minimal* if no proper nonempty subset of the support is also a support. An invariant x is said to be *minimal* if there is no other invariant x' such that $x[t] \leq x'[t]$ for all t. From the point of view of linear programming, the t-invariants of C^1 are the extremal solutions of $C^1x^1 = 0, x^1 \ngeq 0$. As soon as a minimal t-invariant x^1 of N^1 is known, it is easy to reconstitute the corresponding t-invariant x of N as shown in the previous section.

The minimal t-invariants of N^1 are obtained by computing the extremal solutions of $C^1x^1 = 0$ using classical linear programming approaches ([10], [11]). This computation is usually very fast, due to the limited size of C^1. If, for each component, there exists at least one t-invariant in which this component is strictly positive, then N^1, and thus N, is consistent. Furthermore, the minimal t-invariants of N are derived from the t-invariants of N^1 as shown in the previous sub-section (i.e. by firing as many transitions as possible starting from the firing of the input transitions of N as required by the minimal t-invariants of N^1).

4.4 Software Implementation

Software which analyzes a CFIO net is available. It detects whether the net is consistent and provides all minimal t-invariants.

The different steps of the algorithm are summarized as follows:

1. Check and remove the circuits as shown in Result 5. If, in at least one of the circuits, a place has more than one input transition, the net is inconsistent. A message is displayed and the computation stops.
2. Find inequalities (3) starting from the incidence matrix of the initial net N. This is done by applying repetitively relations (2) starting from the output transitions.
3. Find all the independent sets of values (positive or null) which turn inequalities (3) into equalities. This is done by using a classical linear programming approach to find the extremal solutions of $C^1x^1 = 0$, where C^1 is the incidence matrix of the reduced net N^1 derived from inequalities (3).
4. Decide if N is consistent by verifying if each of the components of a transition vector has a positive value in at least one minimal t-invariant.
5. Construct the minimal t-invariants of N as indicated in Section 4.3.

Note that the reduced net appears only through its incidence matrix, which is derived from inequalities (3).

5 Planning of Manufacturing Systems

Planning a manufacturing system consists of determining the list of operations to be performed during some consecutive periods of time. In terms of Petri net models, we have to provide the list of transition firings which occur during these periods. Hereafter, we show how to use CFIOs in order to cope with the computational burden of this task. As described in the following section, this trade-off is achieved at the expense of flexibility. Using CFIOs as the basis to compute the planning guarantees that the qualitative properties which are essential for managing manufacturing systems are preserved.

5.1 Decomposition of a Petri Net

Let us consider a Petri net N with input and output transitions which models the arrival and the departure of parts from the system.
A subset NC of N is a *CFIO of N* if:

1. NC is a connected CFIO net.
2. The input (resp. output) transitions of NC are input (resp. output) transitions of N.
3. Every node (place or transition) of NC is a node of N.
4. For every transition t of NC, the set of input (resp. output) places of t in NC is equal to the set of input (resp. output) places of t in N.

Definition 5: We call a Petri net N with input and output transitions, a *decomposable* Petri net, if there exist consistent CFIOs $NC_1, NC_2, \ldots, NC_r$, of N, such that:

$$N = \bigcup_{i=1}^{r} NC_i$$

Result 7: A decomposable Petri net such that each elementary circuit contains at least one token:

(i) is consistent
(ii) can be kept bounded by controlling its input and output transitions
(iii) is live

Proof:

(i) Let C be the incidence matrix of the decomposable Petri nets N and $NC_1, NC_2, \ldots, NC_r$ a set of consistent CFIOs such that $N = \bigcup_{i=1}^{r} NC_i$.

The consistency of NC_i, $i = 1, 2, \ldots, r$ implies that there exists a vector $x^i \in \mathbb{N}^m$such that $Cx^i = 0$ and:

$$x_k^i \begin{cases} > 0, & \text{if } t_k \in NC_i \\ = 0, & \text{otherwise} \end{cases}$$

Let us consider $x^* \in \mathbb{N}^m$ such that

$$x_k^* = \sum_{i=1}^{r} x_k^i \text{ for } k = 1, 2, \ldots, m \tag{6}$$

then

$$Cx^* = C\sum_{i=1}^{r} x^i = \sum_{i=1}^{r} Cx^i = 0 \tag{7}$$

Furthermore

$$x_k^* = \sum_{i=1}^{r} x_k^i > 0 \text{ for } k = 1, 2, \ldots, m \tag{8}$$

because every transition t_k, $k = 1, 2, \ldots, m$ belongs at least to one of the CFIOs, and thus one of the x_k^i, $k = 1, 2, \ldots, m$ is strictly positive for at least one $i \in \{1, 2, \ldots, r\}$.

From equations (7) and (8) we conclude that N is consistent.

(ii) Given that $NC_1, NC_2, \ldots, NC_r$ are *consistent* subnets, we consider a vector x such that:

$$x = \sum_{i=1}^{r} n_i x^i = [x_1, x_2, \ldots, x_m] \tag{9}$$

where x^i are the vectors introduced in *(i)* above, and n_i are non-negative integers. Then if transition t_k is fired x_k times, $k = 1, 2, \ldots, m$ (which is always possible) the resulting marking is the initial marking.

Thus, N can be kept bounded if the transitions are fired according to (9).

(iii) Liveness results from the fact that N is the union of CFIO nets.

Q.E.D.

5.2 Manageability of a manufacturing system

As far as manufacturing systems are concerned, the goal is to satisfy a variable demand (i.e. firing the output transitions a given number of times) while optimizing some secondary criteria, such as minimizing work-in-process or balancing the utilization of the machines.

We assume that the Petri net model N of the manufacturing system at hand is decomposable. Let $NC_1, NC_2, \ldots, NC_r$ be some consistent CFIOs such that $N = \bigcup_{i=1}^{r} NC_i$. For each NC_i, there exists at least one integer vector $x^i > 0$ as defined in result 7. We select a vector x^i whose elements have a highest common factor equal to 1. Usually x^i is not unique, but we decide to select only one x^i for each NC_i.

Activating NC_i consists in firing the transitions of NC_i as many times as required by the value of the corresponding component in x^i. Several firing orders are usually available. In fact, any enabled transition can be fired at any time unless the number of times this transition has already been fired is equal to the corresponding component of x^i.

Note:

(i) Let $M \in R(M_0)$ such that M is reachable by sequential activation of the CFIOs of N. Then, if M_0 is such that any elementary circuit contains at least one token, it is possible to reach M_0 from M by sequential activations of the same CFIOs. This is because these CFIOs are reversible for the subset of M_0 corresponding to the CFIOs (see Result 4).

(ii) If the CFIOs corresponding to N are all the CFIOs built from the minimal t-invariants, then N is reversible for any marking M_0 such that each elementary circuit contains at least one token.

According to Result 7, it is clear that, if we control the system by sequential activation of the CFIO nets of the Petri net model, we are sure that the system will retain its expected qualitative properties, namely boundedness, reversibility, and liveness.

However, we have to emphasize that choosing only one x^i vector for each NC_i reduces the number of possible ways NC_i can be activated and thus the number of possible ways the whole system can be controlled. In fact, this approach is a trade-off between flexibility of the control and the computational burden. Maximal flexibility of the control would be obtained by using the minimal CFIOs of the Petri net model, i.e., those CFIOs whose transitions correspond to the *minimal* t-invariants.

The number of times each CFIO has to be activated to satisfy the demand may also depend on the secondary criteria introduced previously, since there are usually several ways to meet the demand.

Finally, note that the number of times each CFIO has to be activated provides the number of times each transition has to be fired to meet the demand in a way which retains the expected qualitative properties. The result of this step is the production plan. Defining the order is the scheduling activity. The order of these firings may depend on other constraints, like resource constraints. A planning activity based on CFIOs is illustrated in the following example.

5.3 Example

We consider two product types whose manufacturing processes are represented in Figure 7. In each box we show the machines on which the corresponding operation can be performed. The manufacturing times are given in parentheses.

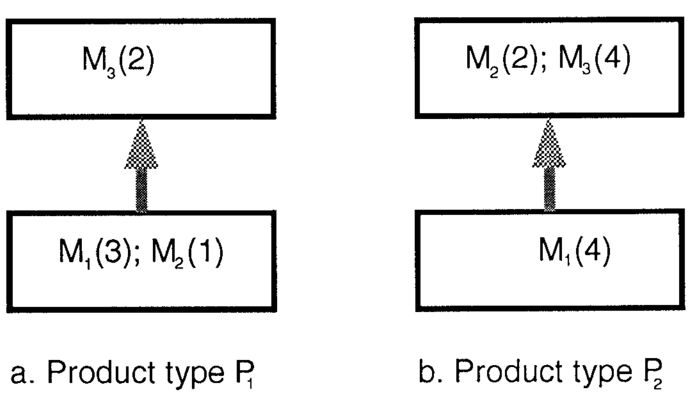

Figure 7: Two manufacturing processes

The Petri net model of these processes are given in Figure 8.

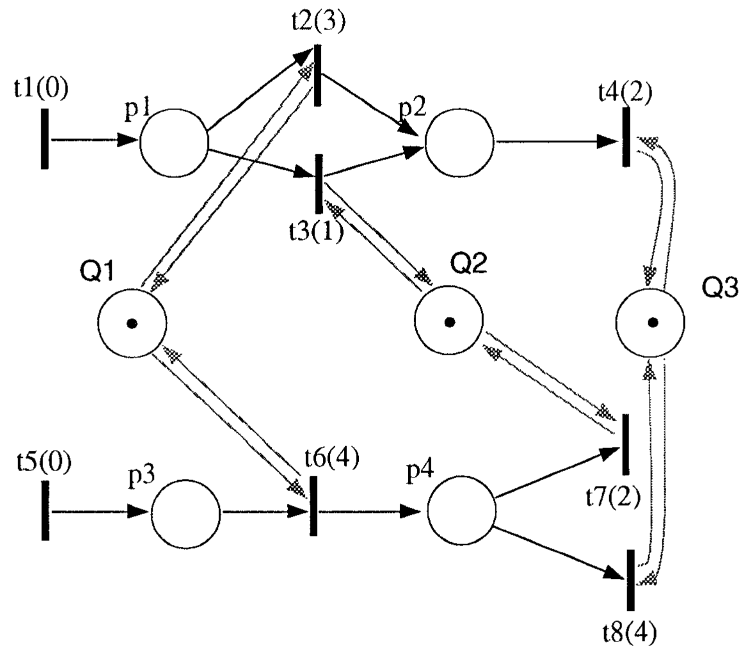

Figure 8: Petri net model

The unique token in places Q_1, Q_2, Q_3 are used to ensure that one machine cannot be used to manufacture more than one product at the same time.

In this example, we construct the following consistent CFIOs: (we do not consider the self-loops)

$$NC_1 =< t_1, p_1, t_2, p_2, t_4 >$$
$$NC_2 =< t_1, p_1, t_3, p_2, t_4 >$$
$$NC_3 =< t_5, p_3, t_6, p_4, t_7 >$$
$$NC_4 =< t_5, p_3, t_6, p_4, t_8 >$$

and the corresponding t-invariants are:

$$H_1 =< 1,1,0,1,0,0,0,0 >$$
$$H_2 =< 1,0,1,1,0,0,0,0 >$$
$$H_3 =< 0,0,0,0,1,1,1,0 >$$
$$H_4 =< 0,0,0,0,1,1,0,1 >$$

Assume that the demand is given for 4 consecutive periods as shown in Table 1. The costs are presented in Table 2.

Table 1 : Demand

Period / Product	1	2	3	4
1	2	15	7	3
2	6	1	4	17

Table 2 : Costs

Product	Backlogging Cost	Inventory Cost
1	20	4
2	30	2

Assume that the length of an elementary period is 40. We denote by $a_{i,k}$ the number of times the i^{th} CFIO has to be activated during the k^{th} period.

The capacity constraints are as follows:

$$\begin{array}{lllllll} 4a_{1,k} & & & + & 4a_{3,k} & + & 4a_{4,k} & \leq 40 \\ & & a_{2,k} & + & 2a_{3,k} & & & \leq 40 \\ 2a_{1,k} & + & 2a_{2,k} & + & 4a_{4,k} & & & \leq 40 \end{array} \tag{10}$$

The left hand side of each inequality represents the utilization time of the related machine.

For this example the criterion consists of minimizing the sum of the inventory and backlogging costs. The criterion to be minimized can be written as follows:

$$\mathcal{F} = \sum_{r=1}^{4} \left\{ 4\left[\sum_{k=1}^{r}(a_{1,k} + a_{2,k} - d_k^1)\right]^+ + 20\left[\sum_{k=1}^{r}(d_k^1 - a_{1,k} - a_{2,k})\right]^+ + \right.$$
$$\left. + 2\left[\sum_{k=1}^{r}(a_{3,k} + a_{4,k} - d_k^2)\right]^+ + 30\left[\sum_{k=1}^{r}(d_k^2 - a_{3,k} - a_{4,k})\right]^+ \right\}$$

where d_k^i is the demand for product i during period k.

By setting

$$\begin{aligned} y_r^1 &\geq \sum_{k=1}^{r}(a_{1,k} + a_{2,k} - d_k^1) \\ z_r^1 &\geq \sum_{k=1}^{r}(d_k^1 - a_{1,k} - a_{2,k}) \\ y_r^2 &\geq \sum_{k=1}^{r}(a_{3,k} + a_{4,k} - d_k^2) \\ z_r^1 &\geq \sum_{k=1}^{r}(d_k^2 - a_{3,k} - a_{4,k}) \end{aligned} \tag{11}$$

the problem to solve becomes:

$$\text{Minimize}: \sum_{r=1}^{4}(4y_r^1 + 20z_r^1 + 2y_r^2 + 30z_r^2)$$

subject to (10), (11) and

$$a_{i,k} \geq 0 \quad i = 1,2,3,4; \ k = 1,2,3,4$$
$$y_r^s \geq 0, \ z_r^s \geq 0 \quad r = 1,2,3,4; \ s = 1,2$$

The result is the number of times each transition has to be fired during each elementary period. (See Table 3)

Table 3 : Planning Results: transition firings

Transition / Period	t1	t2	t3	t4	t5	t6	t7	t8
1	2	2	0	2	6	6	6	0
2	15	10	5	15	2	2	2	0
3	7	0	7	7	10	10	10	0
4	3	0	3	3	10	10	10	0

From Table 3 we derive the production during each elementary period. (See Table 4)

Table 4 : Production quantities

Product / Periods	P1	P2
1	2	6
2	15	2
3	7	10
4	3	10

The subsequent step would be to define the starting time of each transition firing. This constitutes the scheduling problem, which is NP-hard, and beyond the scope of this paper. Nevertheless, it must be noted that the results of planning cannot always be implemented in a schedule. In that case, the solution is to recompute a new planning with more stringent capacity constraints.

6 Conclusion

The contribution of the paper can be summarized as follows: (i) introduction of a special class of conflict-free Petri nets (CFIOs) to the literature which makes possible modeling and analysis of non-cyclic discrete systems, (ii) theoretical results related to properties of CFIO nets, (iii) reduction rules for simplified computation of the t-invariants of CFIOs, (iv) decomposition of a complex model of the overall system into a set of simple CFIOs and the use of these CFIOs for the overall system control/management, and (v) application of the above methodology to perform production planning in manufacturing systems.

Finally, the most important contribution of this methodology is the development of a tool for both behavior evaluation as well as management of non-cyclic discrete systems. If the system is managed using this tool, the qualitative aspects can be ignored during the computation of control because these aspects are built into the model. Furthermore, it provides an approach to achieve an appropriate compromise (or balance) between the flexibility of the system and the computational burden during management. We hope that the motivation for manufacturing systems has been well demonstrated via a realistic example, small in size for the sake of brevity. The next step of our research is to study further the Petri net model structure to develop new scheduling approaches.

Bibliography

1. P. Chretienne, "Les Réseaux de Petri temporisés," Thèse d'Etat, Université Paris VI, France, 1983.
2. G.Harhalakis, J.-M. Proth, "Some other Problems in the Design and Use of Modern Production Systems", Journal of Applied Stochastic Models and Data Analysis, vol.7, no 1, 1991, pp 33–45.
3. H.P. Hillion, and J.M. Proth, " Performance Evaluation of Job-Shop Systems using Timed Event Graphs," *IEEE Transaction on Automatic Control*, vol. 34, no. 1, pp. 3–9, 1989.
4. L.E. Holloway, and B.H. Krogh, "Synthesis of Feedback Control Logic for a Class of Controlled Petri Nets," *IEEE Transaction on Automatic Control*, vol. 35, no. 5, May 1990.
5. S. Laftit, J.M. Proth, and X.L. Xie, "Optimization of Invariant Criteria for Event Graphs," *IEEE Transaction on Automatic Control*, vol. 37, no. 5, pp. 547–555, May 1992.
6. T. Murata, "Petri nets: Properties, Analysis, and Applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, April 1989.
7. J.M. Proth, and F. Vernadat, " Discrete Manufacturing Systems: from specification to evaluation," *The Second International Conference on Automation Technology, Honorary Volume (Keynote and Invited Speakers)*, Research Center for Automation Technology, Yuan-Ze Institute of Technology, Chungli, TAIWAN, R.O.C, 1992.
8. C.V. Ramamoorthy, and G.S. Ho, "Performance Evaluation of Asynchronous Concurrent Systems using Petri nets," *IEEE Transaction on Software Engineering*, vol. SE-6, no. 5, pp. 440–449, 1980.
9. C. Ramchandani, "Analysis of Asynchronous Concurrent Systems by Timed Petri Nets," *Tech. Rep. 120*, Lab. Computer Sci., Mass. Inst. Techno., Cambridge, MA, 1974.
10. K. Lautenbach, "Linear Algebraic Techniques for PT Nets," *Lecture Notes in Computer Science — Advanced in Petri Nets 1986*, vol. 1, pp. 142, 1986.
11. T.C. Hu, *Integer Programming and Networks Flow*. Mass.: Addison Wesley Reading, 1969.
12. G. Memmi, "Linear Algebra in Net Theory," *Lecture Notes in Computer Science — Net Theory and Applications*, pp. 213–223, 1979.
13. J. Sifakis, "Structural Properties of Petri Nets," *Lecture Notes in Computer Science — Mathematical Foundations of Computer Science*, pp. 474–483, 1978.
14. F. Dicesare, G. Harhalakis, J.-M. Proth, M.Silva, F.B.Vernadat eds. *Practice of Petri Nets in Manufacturing*. London: Chapman and Hall, 1993.