

# THESIS REPORT

Master's Degree

## A New Class of Petri Nets for Modeling, Planning and Scheduling of Flexible Manufacturing Systems

*by M.M. Leventopoulos*

*Advisors: G. Harhalakis*

*I. Minis*

*J-M. Proth*

M.S. 94-10



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

**A NEW CLASS OF PETRI NETS FOR MODELING, PLANNING  
AND SCHEDULING OF FLEXIBLE  
MANUFACTURING SYSTEMS**

**by**

**Marios Markos Leventopoulos**

Thesis submitted to the Faculty of the Graduate School  
of The University of Maryland in partial fulfillment  
of the requirements for the degree of  
Master of Science  
1994

**Advisory Committee:**

Assistant Professor Ioannis Minis, Chairman/Co-advisor  
Associate Professor Michael Fu  
Associate Professor Mark Shayman



## **ABSTRACT**

**Title of Thesis: A NEW CLASS OF PETRI NETS FOR MODELING, PLANNING  
AND SCHEDULING OF FLEXIBLE MANUFACTURING SYS-  
TEMS**

**Name of degree candidate: Marios Markos Leventopoulos**

**Degree and Year: Master of Science, 1994**

**Thesis directed by: Associate Professor George Harhalakis, Institute for Systems**

**Research**

**Assistant Professor Ioannis Minis, Institute for Systems Research**

**Dr. Jean-Marie Proth, INRIA Metz and Institute for Systems**

**Research**

This work addresses the modeling, verification, planning and scheduling problems of non-cyclic discrete systems with emphasis on Flexible Manufacturing Systems. We introduce a special type of Petri nets, the Conflict-Free nets with Input and Output transitions (CFIO nets) that provide a much needed platform on which all these problems can be tackled in a unified manner. It is shown that CFIO nets are live, reversible, if consistent, and can be kept bounded under certain conditions. We develop reduction rules which facilitate the detection of the above properties. We then take advantage of the qualitative properties of CFIO nets and use those models to develop a linear programming formulation of the production planning problem. Finally we use the CFIO nets, along with the production planning results, to develop the production schedule of the Flexible Manufacturing system.



## ACKNOWLEDGMENTS

I would like to thank the following people:

A special thanks to the late Dr. George Harhalakis, without whose support and guidance this thesis would not have been possible;

Dr. Jean-Marie Proth, whose vast experience in the manufacturing industry and whose theoretical background contributed immensely to the formation of this thesis;

Dr. Ioannis Minis, who worked closely with me and dedicated much time and effort to the completion of this thesis;

Dr. Chang-Pin Lin and Dr. Rakesh Nagi, on whose theoretical work my thesis is founded;

Dr. Jeffrey W. Herrmann, whose erudite comments have greatly improved this thesis;

Dr. Michael Fu and Dr. Mark Shayman for serving on my advisory committee and offering thoughtful comments;

Ms. Mattie Riley, who has patiently helped me with the many administrative details throughout my graduate studies;

The members of the Cim lab — Anshu, Arun, Azhar, Champa, George, Giang, Tom, and Vijay— all of whom I am happy to have worked with and whose company in and outside the lab has made these past three years bearable during difficult times. Their sense of humor has kept me smiling. Thanks guys!

Finally, and most importantly, I would like to thank my beloved partner-in-life CC, for her constant support throughout this endeavour. She made it all worthwhile!



# TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
List of Tables . . . . .	v
List of Figures . . . . .	vi
Chapter 1          Introduction . . . . .	1
Flexible Manufacturing Systems: An Overview . . . . .	1
Brief Review of Petri Nets . . . . .	3
Brief Review of Production Planning and Scheduling . . . . .	4
Thesis Objectives and Structure . . . . .	6
Chapter 2          Background: Petri Nets . . . . .	8
Black and White Petri Nets . . . . .	8
Timed Petri Nets . . . . .	14
Control Places . . . . .	15
Chapter Summary . . . . .	15
Chapter 3          Conflict Free Nets with Input and Output Transitions	17
Definition of CFIO Nets . . . . .	17
Properties of CFIO Nets . . . . .	19
Detection of Properties of CFIO Nets . . . . .	26
Decomposition of a Petri Net Model of a Manufacturing System . . . .	39
Chapter Summary . . . . .	42



Chapter 4	Application of CFIO Nets to Flexible Manufacturing	
	Systems . . . . .	43
	Petri Net Models of Flexible Manufacturing Systems . . . . .	43
	Design Evaluation . . . . .	47
	Planning . . . . .	53
	Scheduling . . . . .	64
	Software Implementation . . . . .	82
	Chapter Summary . . . . .	85
Chapter 5	Conclusions . . . . .	86
Chapter 6	Bibliography . . . . .	88

## LIST OF TABLES

<u>Number</u>		<u>Page</u>
1.	Transition to operation relation for CFIO nets in Figure 13 . . . . .	45
2.	Description of places for the CFIO nets in Figure 13 . . . . .	45
3.	Demand of parts $P_A$ and $P_B$ . . . . .	59
4.	Costs for parts $P_A$ and $P_B$ . . . . .	59
5.	Linear program solution: The number of times each CFIO needs to be activated during each time period . . . . .	61
6.	Planning Results: transition firings for each period . . . . .	63
7.	Production plan . . . . .	63
8.	Production quantities . . . . .	64
9.	Transition firings for the second period . . . . .	74
10.	Transition information for example B . . . . .	79

## LIST OF FIGURES

<u>Number</u>	<u>Page</u>
1. A marked Petri net . . . . .	9
2. Example of a conflict . . . . .	11
3. $S1$ is a siphon: $\bullet S1 \subset S1^\bullet$ , $S2$ is a trap: $S1^\bullet \subset \bullet S1$ . . . . .	11
4. $M = [0, 0, 0, 1]$ is not reachable . . . . .	12
5. Behavior of a control place . . . . .	16
6. A CFIO net . . . . .	19
8. Examples of inconsistent CFIO nets . . . . .	28
7. Circuit Reduction . . . . .	29
9. Circuit reduction rule . . . . .	30
10. Initial net $N$ . . . . .	37
11. Reduced net $N^1$ . . . . .	38
12. Manufacturing operation sequences for parts $P_A$ and $P_B$ . . . . .	44
13. CFIO nets corresponding to the production routings of parts $P_A$ , $P_B$ . . . . .	46
14. Petri net model of FMS including resource constraining places $q_1, \dots, q_6$ . . . . .	47
15. CFIO net $NC_B^1$ modeling the redesigned routing $R_B^1$ . . . . .	50
16. CFIO net $NC_B^1$ after circuit reduction has been performed . . . . .	50

17.	Final version of $NC_B^1$ . . . . .	51
18.	Final version of $NC_B^2$ . . . . .	52
19.	Model of manufacturing processes and resource constraints . . .	53
20.	Modeling of RU type decisions . . . . .	65
21.	Modeling of PS type decisions . . . . .	67
22.	Manufacturing operations for the three part types of example B. .	77
23.	Manufacturing process model. . . . .	78
24.	Scheduling results . . . . .	83



# 1 Introduction

## 1.1 Flexible Manufacturing Systems: An Overview

During the past three decades industrialized countries have made significant changes in manufacturing practices. 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), competitiveness has become global, and unpredictable demands are progressively replacing steady demands.

Factory automation (FA) is one promising way to address these changes [46]. Factory automation is the synthesis of flexible manufacturing systems (FMS), computer-aided design/computer-aided manufacturing (CAD/CAM), and office automation (OA). Among these three basic elements, FMS form the core of FA on the shop floor: they are designed to fill the gap between high-production transfer lines and low-production NC machines. A flexible manufacturing system is capable of producing a wide variety of products (or parts) with virtually no time lost for change from one product to the next [47]. To achieve this objective FMS integrate industrial robots, numerical control (NC) machines, and automated material-handling systems into a manufacturing cell. In view of the capital-intensive nature of FMS it is important to design and operate them as efficiently as possible.

The problems related to FMS can be classified into design and operational ones. Design problems are concerned with the optimal selection of the FMS components and their interconnection, including [48]: (i) selection of part families, (ii) selection of pallets and fixtures, (iii) selection of material-handling systems, (iv) selection of the

information system that links the various FMS modules and the FMS with the rest of the FA modules, (v) layout of the FMS. Operational problems relate to the utilization of the FMS and include [49]: (i) batch sizing, (ii) balancing of the workload, (iii) long and medium term planning, (iv) scheduling and dispatching, (v) tool management, (vi) response to changes in demand, (vii) reaction to resource disruptions.

The decisions involved in the design and operation stages of FMS are complex. Thus it is necessary to provide the decision makers with aids that will allow them to plan, design and operate the FMS as effectively as possible. Such critical aids include methods for system modeling and analysis which support the selection of “good” design alternatives and operational policies. Furthermore a *unified* environment that supports modeling, verification and performance evaluation is highly desirable.

Since FMS are discrete-event, dynamic systems, many tools exist to model and analyze them. For instance, queueing theory, state-transition analysis, mathematical programming, and simulation may be used for analyzing the system performance, while entity-relationship approaches and the CIMOSA<sup>1</sup> related tools are used for the functional specification of such systems.

In this thesis we develop Petri net-based models to support the modeling and verification of FMS designs, as well as to support short term planning and scheduling during FMS operation. The following subsections review relevant work in Petri nets. Significant work in Production Planning and Scheduling is also reviewed to provide a background for the applications addressed by the proposed tools.

---

<sup>1</sup> Computer Integrated Manufacturing — Open System Architecture, ESPRIT program

## 1.2 Brief Review of Petri Nets

Petri nets have been successfully used in the specification and the functional modeling of manufacturing systems. The reader may refer to Silva and Valette [44], and Murata [40] for an introduction in the area of Petri nets applied to flexible manufacturing. DiCesare *et al.* [33] present a thorough analysis of the subject. Timed Petri Nets, a variation of Petri nets that incorporates time information, have been extensively used for performance analysis of manufacturing systems. Murata [40] gives a good introduction on timed Petri nets. For use of timed Petri nets in manufacturing systems see also Di Mascolo *et al.* [38], Hillion *et al.* [39], and Ramchandani [42].

Manufacturing systems are usually classified in two types by Petri net researchers: (i) *Cyclic production* manufacturing systems, in which part-production rates are constant. (ii) *Non-cyclic* manufacturing systems in which part-production rates vary with time. The first type of systems is common in a make-to-stock environment, while the second type is common in a make-to-order environment. Numerous analytical techniques are available to support the preliminary design of cyclic manufacturing systems. Such systems are typically modeled using strongly connected event graphs, a special class of conflict-free Petri nets the analytical properties of which are particularly powerful. Desrochers [25] and Silva [26],[44] present a survey of event graphs and their use in manufacturing systems. Rozenberg [31] has studied the use and application of event graphs. For a more extensive study on event graph-related properties see Berthelot [28] and Rozenberg [29], and for event graph invariant analysis see Silva [26], [30]. Event graphs are also used for performance evaluation [39], [41] and optimization of cyclic systems [2].



For the more general (non-cyclic) FMS, as well as for job-shops, which are also non-cyclic, strongly connected event graphs techniques are not appropriate; this is because such systems are not conflict-free. In addition, one must introduce and remove parts from the system and this process cannot be represented by a strongly connected net. This motivates the quest for more powerful modeling tools than event graphs.

### **1.3 Brief Review of Production Planning and Scheduling**

Planning a manufacturing system consists of determining the type and number of parts to manufacture in a given time horizon, in order to satisfy a given demand. Given the production plan, the central concern of job shop scheduling is to allocate the necessary production tasks to limited resources and determine the task starting times, in order to optimize a given set of criteria while satisfying certain manufacturing constraints.

A substantial body of literature addresses different aspects of planning and scheduling, including machine sequencing (Baker [9]), mathematical modeling of the general resource-constrained project scheduling problem (Rodammer [23]), mixed model assembly line scheduling (Milteneburg *et al.* [19], and [20]), network programming methods (Zahorik *et al.* [24]) and job-shop scheduling (Adams [8]).

The vast literature in the scheduling field has been reviewed and classified by a number of authors. Panwalkar and Iskander [22] present and critique a comprehensive list of various priority rules proposed in the literature. Rodammer and White [23] survey various approaches to the production scheduling problem, such as machine sequencing, resource-constrained project scheduling, discrete event simulation and

artificial intelligence methods. MacCarthy and Liu [12] and Lawler *et al.* [18] have tabulated the types of scheduling problems addressed in the literature along with the most important solution methods.

One of the important objectives for the n-job, m-machine, job shop problem is the minimization of the production makespan. Various approaches have been proposed for this problem, including mixed integer linear programming (Lageweg *et al.* [17]), branch-and-bound searches (Greenbreg [13], Barker *et al.* [10] and Carlier *et al.* [11]) and several heuristic methods (Adams *et al.* [8] and Stoner *et al.* [45]). However, optimal methods are computationally prohibitive for practical cases, given that the underlying problem is NP-hard<sup>2</sup>. A detailed classification and complexity analysis of scheduling problems can be found in Lawler *et al.* [18] and Herrmann *et al.* [14]. There has been, as mentioned, much research on single machine and multiple machine scheduling problems, but little work has been performed on systems with *routing flexibility* since linear programming models of the scheduling problem become very complex. Hoitomt [15], [16] and Owens [21] have adapted Lagrangian relaxation techniques to decompose complex scheduling problems, but there is no support for shared resources.

The actual generation of schedules has not received much attention in the Petri net community. Carlier and Chretienne [32] developed a polynomial algorithm, which, given a firable sequence for the underlying model, computes the earliest feasible schedule for that sequence. Shih and Sekiguchi [43] use Petri nets to simulate, through a beam search, the evolution of an FMS. Lee and DiCesare [37] introduced a search

---

<sup>2</sup> For a guide to the theory of NP-completeness the reader is referred to Garey and Johnson [34]

algorithm that operates on the reachability tree of a Petri net model of the manufacturing system to produce a feasible schedule. Their method supports flexible routings and shared resources. Furthermore, it avoids the size explosion of the reachability tree by using a heuristic that disregards parts of the tree on the basis of an approximate evaluation.

It is important to note at this point that problems arise in the integration of research done with Petri nets in the different areas of manufacturing systems modeling and analysis. A comprehensive approach needs to address specification, functional modeling, preservation of qualitative properties, planning, and scheduling in a unified manner. To date research conducted in these areas has not employed the same type of Petri nets. For example, different studies have used Color Petri nets, simple Petri nets (also called black & white Petri nets) and event graphs.

## **1.4 Thesis Objectives and Structure**

This thesis addresses the modeling and verification problems of FMS and non-cyclic manufacturing systems during system design as well as the planning and scheduling problems during system operation. We provide a much needed platform on which all these problems can be tackled in a unified manner. Thus, we eliminate the need for system designers to create different models for system representation, qualitative analysis, and planning and scheduling.

For this purpose we introduce the Conflict Free nets with Input and Output transitions (CFIOs), a subset of simple Petri nets. CFIOs offer strong graphical power for non-cyclic manufacturing systems. We show that modeling realistic manufacturing

systems with Petri nets decomposable to consistent CFIOs guarantees the desired qualitative aspects of the system, i.e. the absence of deadlocks and overflows, the ability of the system to recover from errors, the ability to reach all the states for which it was designed, and the presence of certain mutual exclusions in the use of shared resources. An algorithm that tests a CFIO for such important properties is presented. Finally we use these models for planning and scheduling. For these problems we provide powerful heuristics that use aggregation techniques to find a good approximation to the optimal solution.

The remainder of this work is organized as follows: In Chapter 2 we provide critical definitions and properties of Petri nets. In Chapter 3 the CFIO nets are defined and the notion of *manageable* manufacturing systems is introduced. We analyze the qualitative properties of CFIO nets and relate them to manageability. These properties are employed in Section 4.3, to perform planning. A small example illustrates the proposed procedures. In Section 4.4 we introduce a critical path-based heuristic that computes a schedule from the planning information derived in Chapter 4.3, such that the makespan of production is within a given time period. An example is also presented to illustrate this procedure. In Chapter 5 we present our conclusions.

## 2 Background: Petri Nets

This chapter gives a brief description of important types of Petri nets. In Section 2.1 we introduce black-and-white Petri nets; we use the definition and notation of Murata [40] and DiCesare *et al* [33]. In Section 2.2 we discuss timed Petri nets and in 2.3 we define the control places which are used for modeling external decisions.

### 2.1 Black and White Petri Nets

The structure 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).

**Definition 1:** A Petri net is a quadruple  $N = \langle P, T, F, W \rangle$  where :

- $P = \{p_1, p_2, \dots, p_n\}$  is a finite set of places,
- $T = \{t_1, t_2, \dots, t_m\}$  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),
- $W : F \rightarrow \mathbb{N}^+$ , where  $\mathbb{N}^+$  is the set of positive natural numbers, is a weight function which assigns a strictly positive value to each arc.

**Definition 2:** A Petri net is said to be *ordinary* if all its arc weights are equal to one.

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

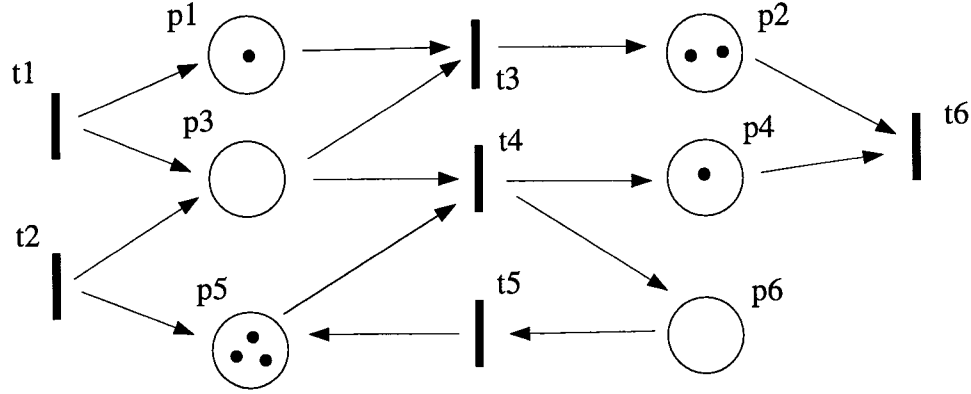


Figure 1: A marked Petri net

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

**Definition 3:** A Marked Petri net is a Petri net  $PN = \langle N, M \rangle$  where:

- $N = \langle P, T, F, W \rangle$  is a Petri net as defined above,
- $M : P \rightarrow \mathbf{N}$  is called a marking of  $PN$ , where  $\mathbf{N}$  is the set of natural numbers.

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; its tokens are represented by dots. The marking of this net is  $M = \langle 1, 2, 0, 1, 3, 0 \rangle$ , i.e.,  $M(p_1) = 1$ ,  $M(p_2) = 2$  and so on.

**Definition 4:** A Petri net  $N = \langle P, T, F, W \rangle$  is called *pure* if  $\bullet t \cap t^\bullet = \emptyset$ ,  $\forall t \in T$ .

A pure Petri net can be represented by an incidence matrix

$$C = [c_{i,j}], \quad i = 1, 2, \dots, n; j = 1, 2, \dots, m$$

where

$$c_{i,j} = \begin{cases} -W(p_i, t_j), & \text{if } p_i \in \bullet t_j \\ +W(p_i, t_j), & \text{if } p_i \in t_j^\bullet \\ 0, & \text{otherwise} \end{cases}$$

As an example, the following incidence matrix corresponds to the ordinary Petri net presented in Figure 1. Each row corresponds to a place, and each column corresponds to a transition.

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

For example, the value  $c_{3,4} = -1$  in the 3<sup>rd</sup> row, 4<sup>th</sup> column, corresponds to the directed arc connecting place  $p_3$  with transition  $t_4$ .  $c_{3,4} = -1$  because  $t_4$  is an output transition of  $p_3$  and the weight of the corresponding arc is one.

When a manufacturing system is modeled using Petri nets, tokens represent parts, resources, or other information, places represent buffers, and transitions represent operations.

In a Petri net, the marking is the state of the net, and it evolves according to the following rules:

### Transition firing rules

1. A transition  $t$  is *enabled* if each input place  $p$  of  $t$  has at least  $W(p, t)$  tokens, where  $W(p, t)$  is the weight of the arc from  $p$  to  $t$ ; that is,  $M(p) \geq W(p, t), \forall p \in \bullet t$
2. A firing of an enabled transition  $t$  removes  $W(p, t)$  tokens from each input place  $p$  of  $t$  and adds  $W(t, q)$  tokens to each output place  $q$  of  $t$ .

3. An enabled transition may or may not fire. That is because the firing of one enabled transition may disable another. For example in Figure 2 although both transitions are enabled, only one can fire. Such conflicts are resolved *non-deterministically*, i.e. the model does not specify how to resolve them.

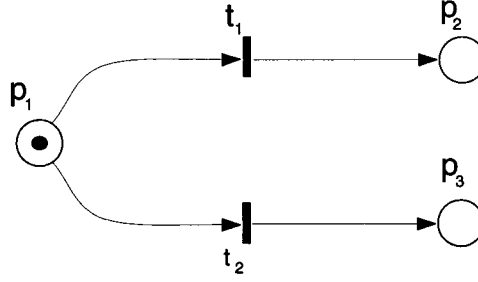


Figure 2: Example of a conflict

**Definition 5:** *Siphon and Trap:* In an ordinary net, a non-empty subset of places  $S$  is called a *siphon* if  ${}^*S \subseteq S^*$ . Once a siphon becomes token-free, it remains token-free. The subset  $S$  is called a *trap* if  $S^* \subseteq {}^*S$ . Once a trap is marked it remains marked.

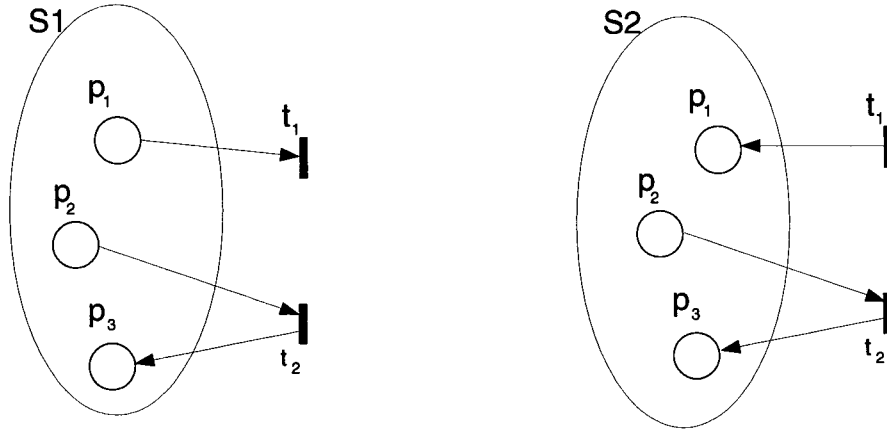


Figure 3:  $S1$  is a siphon:  ${}^*S1 \subseteq S1^*$ ,  $S2$  is a trap:  $S1^* \subseteq {}^*S1$

**Definition 6:** *Reachable marking:* If by firing a sequence  $\sigma$  of transitions, we can reach a marking  $M$  from an initial marking  $M_0$ , we say that  $M$  is *reachable* from  $M_0$ , and that  $\sigma$  is a *feasible firing sequence* (or *firable sequence*) starting at  $M_0$  and



leading to  $M$ . We denote this as follows:

$$M \in R(M_0)$$

and

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

The two markings are related by the Petri net *state equation*:

$$M = M_0 + C\alpha_\sigma \quad (1)$$

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

Note that the state equation by itself provides only a necessary condition for  $M$  to be reachable from  $M_0$  (i.e.  $M \in R(M_0)$ ), as there may be no feasible firing sequence  $\sigma$  corresponding to the counting vector  $\alpha_\sigma$ . Consider for example the following marked Petri net, shown in Figure 4:

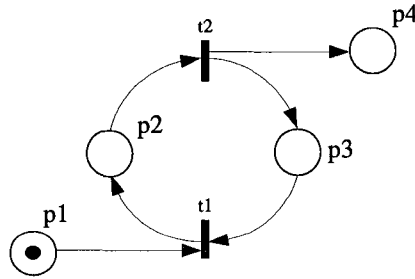


Figure 4:  $M = [0, 0, 0, 1]$  is not reachable

$$C = \begin{matrix} & \begin{matrix} t_1 & t_2 \end{matrix} \\ \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{matrix} & \begin{bmatrix} -1 & 0 \\ 1 & -1 \\ -1 & 1 \\ 0 & 1 \end{bmatrix} \end{matrix}$$

$$M_0 = [1, 0, 0, 0]$$

The marking  $M = [0, 0, 0, 1]$  and the counting vector  $x = [1, 1]^T$  satisfy the state equation. However, there exists no feasible firing sequence such that  $M_0 \xrightarrow{\sigma} M$ , and thus the marking  $M$  is not reachable from  $M_0$ .

**Definition 7:** In the state equation consider the case where  $M = M_0$ . The integer solution of the homogeneous equation  $Ca_\sigma = 0$  is called a *t-invariant*.

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  $t$ .

The following chapters show that t-invariants provide a powerful tool for qualitative analysis of Petri nets, and facilitate the planning and scheduling approach of manufacturing systems via the use of CFIO nets.

**Definition 8:** 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 includes each node only once. For instance,  $\langle p_5, t_4, p_6, t_5, p_5 \rangle$  is an elementary circuit of the net in Figure 1.

**Note:** A directed path which starts and ends at the same node, but does not necessarily include each node only once, is called a *circuit*.

## 2.2 Timed Petri Nets

The representation of time in models of manufacturing systems is essential. There are two ways of introducing time into Petri nets; by associating time with either transitions or places. In the following we will be using the *three phase timed firing* definition of *deterministically transition-timed Petri Nets* [33].

**Definition 9:** A deterministically transition-timed Petri Net (*t-TPN*) is a couple  $\langle N, Z \rangle$  such that:

1.  $N = \langle P, T, F, W \rangle$  is a Petri net as defined above,
2.  $Z : T \rightarrow \mathbf{Q}^+$ , where  $\mathbf{Q}^+$  is the set of non-negative rational numbers.

$Z(t)$  is called the *firing time* of transition  $t$ . The dynamic aspect of a timed Petri net is also defined through its marking (see definition 3).

The firing of a transition  $t$  can be initiated as soon as the transition is enabled and it comprises in three phases:

1. Upon firing, tokens are removed from the input places of the fired transition according to the weight function. Conflicts are resolved in the same manner as in non-timed Petri nets, i.e. non-deterministically.
2. The firing duration is  $Z(t)$  time units.
3. After  $Z(t)$  time units have elapsed, the firing ends and tokens are added to the output places of the transition according to the weight function.

**Non-reentrant TPNs:** In this work we consider *t*-TPNs for which two firings of the same transition cannot overlap. We can enforce such a restriction by adding a place  $q$  to every transition  $t$  in the net, and connect it such that it is both an input and an output of  $t$ ; i.e.  $q \in {}^\bullet t$  and  $q \in t^\bullet$  :  $\langle q, t \rangle$  is a self-loop. The initial marking of each place belonging to a self-loop is one token only. We denote by  $Q = \{q_1, q_2, \dots, q_l\}$  the finite set of those places. Such a timed Petri net is called *non-reentrant*.

## 2.3 Control Places

Krogh *et al.* [1] were the first to propose an extension of Petri nets to account for external decisions that can influence the behavior of the system. This is achieved by introducing control places to the Petri net (represented by a double concentric circle).

A control place  $p$  has the following properties:

1.  ${}^\bullet p = \emptyset$ : A control place has no input transitions.
2.  $M(p) \leq 1$ : There is at most one token in each control place.
3. The initial and subsequent markings of a control place are decided externally to the system and are not changed by the firing of transitions. That is, a token is *not* removed from a marked control place when one of its output transitions fires (see Figure 5). Obviously, unmarked control places effectively disable their output transitions.

## 2.4 Chapter Summary

In this chapter we have briefly introduced and defined Petri net models that are important for the remainder of this work. The static aspect of the Petri net comprises a bipartite graph, while the dynamic aspect is provided by the net marking. For pure

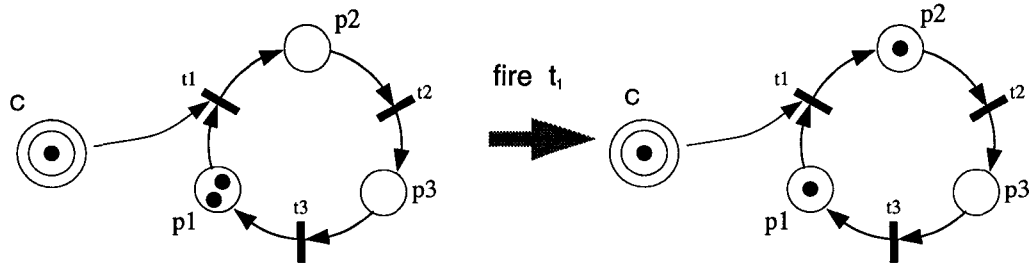


Figure 5: Behavior of a control place

Petri nets the incidence matrix provides an equivalent mathematical representation of the static aspect, and the state equation [Equation (1)] defines the dynamics of the net. Transition timed Petri nets (hereafter simply called *TPNs*) associate time with transition firings. Control places model external decisions.

### 3 Conflict Free Nets with Input and Output Transitions

In this chapter we introduce a new class of Petri nets which are capable of modeling and analyzing non-cyclic manufacturing systems. These nets are called Conflict Free with Input and Output transitions (CFIO) and are defined in Section 3.1. Subsequently, in Section 3.2, we consider the properties of liveness, boundedness, and consistency of CFIOs, and we show their relationship to manageability, a set of properties derived from the need to assess the design of manufacturing systems. Section 3.3 provides methods to evaluate these properties for a given CFIO net. Finally, in Section 3.4 we show that the properties of CFIO nets propagate into Petri nets that are unions of CFIO nets. Such Petri nets are employed in subsequent chapters to model and perform the planning and scheduling of non-cyclic manufacturing systems.

#### 3.1 Definition of CFIO Nets

Conflict Free nets with Input and Output transitions include two types of transitions that support non-cyclic behavior:

- i. **Input transitions** deliver tokens to the net, making it possible to model raw material and component parts that are delivered to the manufacturing system.
- ii. **Output transitions** remove tokens from the net. In manufacturing, these tokens represent finished goods or completed subassemblies leaving the system.

To convert a black-and-white Petri net with input and output transitions to a CFIO, we introduce *control places* that implement external control. Appropriate control turns

such a net into a conflict-free net. As soon as a control decision is applied by means of the control places, some of the transitions are *frozen* (i.e. cannot fire anymore). By removing from the initial Petri net the frozen transitions, the related arcs, and the input and output places that are not connected to non-frozen transitions, we obtain a conflict-free subnet. Such a subnet is referred to hereafter as a Conflict-Free net with Input and Output transitions (CFIO). Note that using this approach a sequence of decisions applied to a non-cyclic manufacturing system can be modeled by a sequential activation of conflict-free subnets of the initial Petri net model.

**Definition 10:** *CFIO nets* form a subclass of ordinary and pure Petri nets with the following three properties:

1. They are structurally *conflict-free*:

$$\bullet t_1 \cap \bullet t_2 = \emptyset, \forall t_1, t_2 \in T$$

This implies that each place has only one output transition.

2. They do not include input and output places (also called source/sink places):

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

3. They include at least one input transition and at least one output transition:

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

$$T - \bullet P \neq \emptyset$$

The net shown in Figure 6 is a CFIO net, where  $t_1$  and  $t_2$  are the input transitions, and  $t_8$  and  $t_9$  are the output transitions.

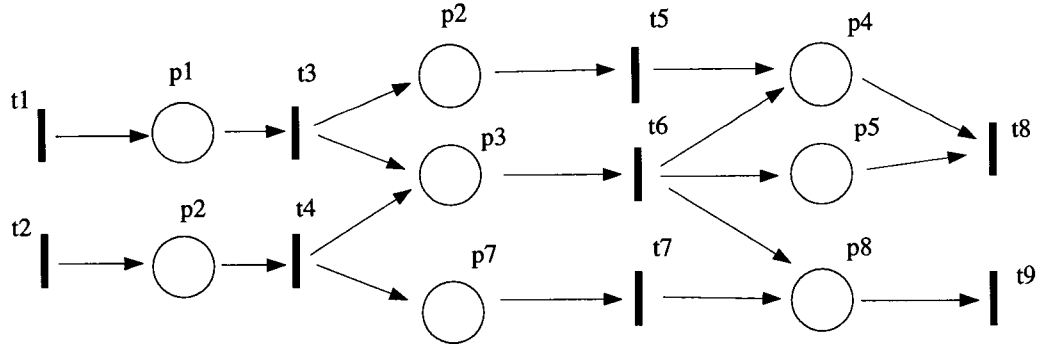


Figure 6: A CFIO net

### 3.2 Properties of CFIO Nets

In order to associate CFIO net properties to important properties of manufacturing systems, we first introduce the notion of *manageability*.

**Definition 11:** A CFIO that models a manufacturing system is manageable if

- i. It is live, i.e., the corresponding manufacturing system is deadlock free.
- ii. It is possible to keep the net bounded i.e. the work-in-process of the corresponding manufacturing system can be held below a given level.
- iii. Any valid state (marking) of the net can be reached from any other valid state. For the manufacturing system, this condition implies that any valid inventory level, as well as any machine availability state, can be reached.

In this section manageability is related to the CFIO properties of liveness, boundedness and consistency. In Section 3.2.1 we show that a CFIO net is structurally live. In Section 3.2.2 we show that although CFIOs are not structurally bounded, CFIO-based models of manufacturing systems can be kept bounded. In Section 3.2.3 we show that reversibility is a consequence of consistency in the case of CFIOs, and that



CFIO nets are reversible for any initial marking provided that all elementary circuits of these nets are marked.

### 3.2.1 Liveness

**Definition 12:** 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  $M_2$ .

Note that if  $N$  is a structurally live Petri net, then there exists a vector  $x$  of positive integers such that  $Cx \geq 0$  (Memmi [6], and Sifakis [7]).

**Result 1:** A CFIO net is structurally live.

To show this result we first prove lemma 1 and subsequently we consider lemma 2 from Murata [40].

**Lemma 1:** Every siphon in a CFIO net contains a circuit.

**Proof:** Let  $N = \langle P, T, F, W \rangle$  be a CFIO net with at least one siphon. Consider any subset  $S$  of  $P$  such that  $S$  forms a siphon, 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 and, 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, it contains a circuit.

**Lemma 2:** A conflict-free net  $N$  is structurally live if and only if every siphon in  $N$  contains a trap [40].

**Proof of result 1:** Since  $N$  is a conflict free net, each place has only one output transition, and therefore the output transitions of places belonging to a circuit also

belong to the circuit. This implies that every circuit in  $N$  is a trap. Furthermore from lemma 1, every siphon in  $N$  contains a circuit. Thus, every siphon in  $N$  contains a trap, and according to lemma 2, it follows that  $N$  is structurally live.

Q.E.D.

Result 1 shows that the liveness property is intrinsic to CFIOs. Thus the first condition of manageability (as defined above) is satisfied for CFIOs.

### 3.2.2 Boundedness

**Definition 13:** 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 Petri net is structurally bounded if and only if there exists a vector  $y$  of positive integers such that  $y^T C \leq 0$  (Memmi [6], and Sifakis [7]). However a CFIO net has at least one input transition, and therefore at least one column of the incidence matrix contains only nonnegative values, of which at least one 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.

**Result 3:** Consider a CFIO net, and assume that the number of firings of all input transitions is bounded. This CFIO net can be kept bounded if and only if the Petri net obtained by removing the input transitions is bounded.

**Proof:**

(a) *Sufficiency:* Let  $A = [A_1 A_2]$  be the incidence matrix of the CFIO net where  $A_1$  comprises the columns of  $T - P^\bullet$  (i.e., the columns that correspond to the input transitions).

Boundedness of the net corresponding to  $A_2$  implies that for any initial marking  $M_0$  there exists an integer  $k_1 > 0$  such that for any  $M \in R(M_0), M(p) \leq k, \forall p \in P$ .

Using the Petri net state equation this can be written as:

$$\begin{aligned} \forall M_0, \exists K_1 = [k_1, k_1, \dots, k_1]^T, \quad k_1 \in \mathbb{N}^+ \text{ s.t.} \\ \forall \alpha_{\sigma_2} \text{ such that } M_0 \xrightarrow{\sigma_2} M, \quad M_0 + A_2 \alpha_{\sigma_2} \leq K_1 \end{aligned} \quad (2)$$

where  $\sigma_2$  is a feasible sequence of transition firings related to  $P^\bullet$  and  $\alpha_{\sigma_2}$  is the counting vector. Let  $\sigma_1$  be a sequence related to the input transitions  $T - P^\bullet$ . Since all input transitions are always enabled it is possible to find a feasible firing sequence  $\sigma_1$ , for any  $\alpha_{\sigma_1}$  and for any initial marking  $M_0$ . Furthermore, since there are no restrictions on how many times each input transition may fire, then for any  $K = [k, k, \dots, k]^T$  with  $k > k_1$ , there exists a feasible sequence of transition firings  $\sigma_1$  such that:

$$A_1 \alpha_{\sigma_1} \leq K - K_1, \quad (3)$$

i.e., we can select  $\alpha_{\sigma_1}$  such that  $A_1 \alpha_{\sigma_1}$  is less than any given positive integer vector.

Equations (2) and (3) above imply that there exists  $K_1$  and  $\sigma_1$  such that:

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

and, thus, the CFIO net can be kept bounded for a given  $\sigma_1$  and for any initial marking  $M_0$ .

(b) *Necessity:* If  $A_2$  as defined above is not bounded, then for any  $K > 0$ :

$$M_0 + A_2\alpha_{\sigma_2} > K$$

for at least one feasible sequence  $\sigma_2$  related to an initial marking  $M_0$ . Note that the columns of  $A_1$  do not contain negative values. Thus,  $A_1\alpha_{\sigma_1}$  is nonnegative for any  $\sigma_1$ . This implies that  $M_0 + A_1\alpha_{\sigma_1} + A_2\alpha_{\sigma_2} > K$ . In other words, for any  $K > 0$  there exists a feasible sequence  $\sigma = \sigma_1 \circ \sigma_2$  such that

$$M_0 + A\alpha_\sigma > K$$

and, consequently, the CFIO net cannot be kept bounded.

Q.E.D.

To relate the above results to manageability consider condition (ii) of definition 11, which requires the ability to maintain the CFIO net bounded. Result 3 provides a way to test for this condition; i.e., if by removing the input transitions the resulting net remains unbounded, then the initial net is also unbounded. This, in turn, implies that the system is non-manageable. The reverse is also true.

We claim that such an unbounded model cannot result from a well-designed manufacturing system. To clarify this point let us consider an unbounded net  $N$  and let  $N_2$  be the net obtained from  $N$  by removing the input transitions. According to Result 3,  $N_2$  is unbounded at some place  $p$ . Thus, for some initial marking, at least one of the inputs of  $p$ , say  $t$ , can fire an infinite number of times. Since there are no

input transitions,  $t$  must drain a token an infinite number of times from a place other than  $p$ . Furthermore, 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 all transitions of which can be enabled an infinite number of times. Furthermore, in this case, all input places of the circuit transitions belong to a circuit; otherwise they would not be able to supply an infinite number of tokens. We have thus shown the following theorem:

**Theorem:** Let  $N$  be a CFIO net and let  $N_2$  be the net obtained by removing all the input transitions of  $N$ . If  $N_2$  is unbounded, then it contains a circuit (not necessarily elementary), all transitions of which have inputs only within the circuit. We will call such a circuit *self-sustained*.

A well-designed manufacturing system should not contain a self-sustained circuit, i.e., it should not be unbounded. The existence of a self-sustained circuit is equivalent with the existence of an uncontrollable manufacturing subsystem, the operation of which cannot be affected by external control. Thus, a CFIO that remains unbounded after removing (or effectively controlling) its input transitions indicates some design or modeling error. Section 3.3.1 shows how to detect such problems that lead to an unmanageable system.

### 3.2.3 Consistency

**Definition 14:** A CFIO net  $N$  is *consistent* if there exists a marking  $M_0$  and a transition firing sequence  $\sigma$  such that  $M_0 \xrightarrow{\sigma} M_0$ , and  $\sigma$  contains every transition of  $N$  at least once.

Sifakis [7] and DiCesare [33] show that consistency in Petri nets holds if and only if there exists a vector  $x > 0$  such that  $Cx = 0$ , where  $C$  is the incidence matrix of the net.

**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).

In the proof we will use the following theorem from Murata [40], which holds for generalized Trap Circuit (TCC) nets, a superclass of CFIO nets:

**Theorem:** In a conflict-free net  $N$ ,  $M_d$  is reachable from  $M_0$  if there exists a nonnegative integer vector  $x = \langle x_1, x_2, \dots, x_m \rangle$  such that  $M_d = M_0 + Cx$ , and every siphon in  $(N_x, M_{0x})$  contains a marked trap; where,  $C$  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 places, output places and their connecting arcs;  $M_{0x}$  denotes the sub-vector of  $M_0$  corresponding to the places of  $N_x$ .

**Proof of result 4:** From Lemma 1, and since all circuits are marked, every siphon in  $N$  contains a marked trap. Since  $M$  is reachable from  $M_0$ , there exists a nonnegative integer vector  $x$  such that:

$$M = M_0 + Cx$$

However, by assumption,  $N$  is consistent and 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 positive constant). Let  $y = z - x$ . We then have  $Cx = -Cy$  and  $M_0 = M + Cy$ .

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

Q.E.D.

**Corollary 1:** For every initial marking  $M_0$  of a consistent CFIO net, such that all elementary circuits contain at least one token each,  $M_1 \in R(M_0)$  and  $M_2 \in R(M_0)$  implies that  $M_1 \in R(M_2)$  and  $M_2 \in R(M_1)$ .

According to this corollary, 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 property of CFIO nets guarantees the third condition of the manageability property of manufacturing systems (see definition 11).

### 3.3 Detection of Properties of CFIO Nets

The previous subsections have established that a consistent CFIO which can be kept bounded satisfies all three manageability conditions. However, since only liveness is intrinsic to CFIOs, in order to assess the manageability of manufacturing systems modeled by CFIOs, it is necessary to develop techniques to determine whether: i) a net can be kept bounded and ii) it is consistent.

In the discussion that followed Result 3 of Section 3.2.2 it was shown that CFIOs that do not contain self-sustained loops can be kept bounded. Thus, to assess this property it suffices to detect the existence of self-sustained circuits in a CFIO net. In the following subsections we introduce a method that i) detects self-sustained circuits in a CFIO net, ii) determines whether the net is consistent, and iii) calculates the t-invariant information, which is necessary for production planning. This method reduces the CFIO net, preserving the consistency property and the t-invariant information, and comprises two basic steps: *circuit reduction* and *path reduction*. These two steps are discussed below.

### 3.3.1 Circuit Reduction

The first step of CFIO net reduction is the *circuit reduction* during which all self-sustained loops (circuits) are detected. If no such circuits are found, then, according to Section 3.2.2, the net under consideration can be kept bounded. Some cases of inconsistency that are caused by a particular circuit formation are also detected during circuit reduction. If no such pathological circuit formations exist, then each elementary circuit of the net is reduced to a single transition. It will be shown that the resulting net maintains the consistency property (or the lack of it) of the original net.

We follow two steps during circuit reduction. First all circuits in the CFIO net are detected. We use an algorithm developed by Knuth [36] that sorts and detects circuits in directed bipartite graphs, such as Petri nets. Secondly we employ Result 5 below, which shows that if there exists a circuit with a place that has more than one input, the net is not consistent. For each circuit detected, it is straightforward to check whether its places have only one input transition. If that is the case, the circuit is reduced to



a single transition. To transform the incidence matrix of the net accordingly, all the columns corresponding to the circuit transitions are replaced by a new column which is the sum of the removed columns. In effect the new transition “inherits” all the input and output places of the reduced circuit. On the other hand, if the circuit has a place with multiple input transitions, then, as shown in the following result, the net is not consistent. This reduction is repeated for all circuits. If no circuits with multiple input places are found, the circuit reduction step terminates successfully. Otherwise it is reported that the net is not consistent.

Result 6 shows that circuit reduction preserves the consistency property. Note that if the reduced circuit was a self-sustained circuit, it would have no input transitions and the resulting transition will be an input transition.

The flowchart of the circuit reduction algorithm is shown in Figure 7.

**Result 5:** Let  $N$  be a CFIO net. 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.

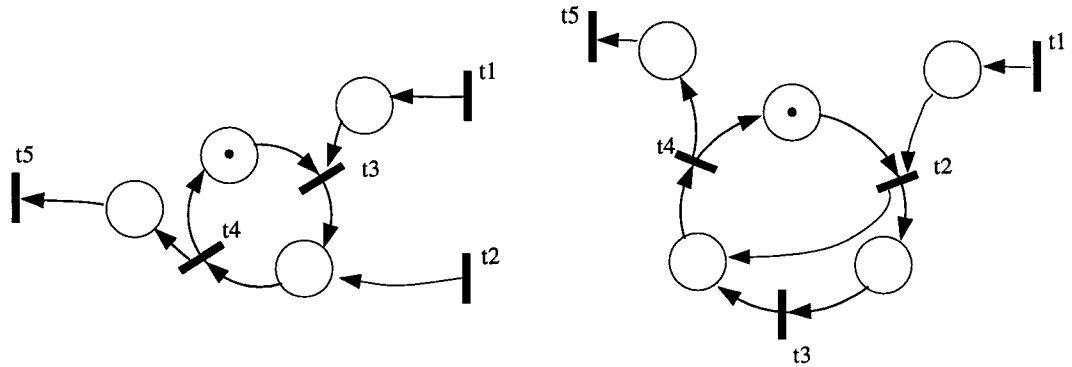


Figure 8: Examples of inconsistent CFIO nets

**Proof of Result 5:** Let  $\gamma \subseteq N$ , be an elementary circuit of  $N$ . Consider any marking  $M_0$  and any transition firing sequence  $\sigma$  which contains every transition at least once.

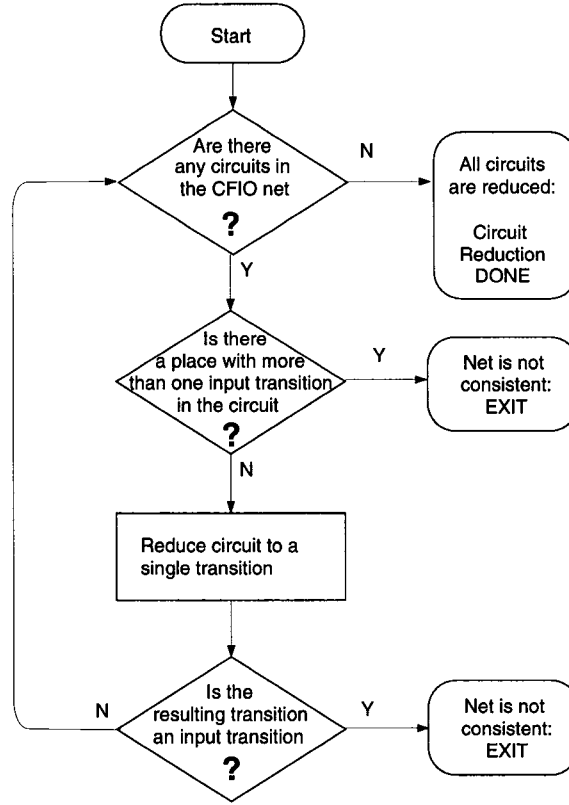


Figure 7: Circuit Reduction

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  $\gamma$  cannot decrease. Now if there exists  $p \in \gamma$  such that  $|\bullet p| > 1$ , firing once every transition  $t \in \bullet p$  will increase the number of tokens in  $\gamma$  by  $|\bullet p| - 1$ . Since  $\sigma$  contains each  $t \in \bullet p$  at least once, firing  $\sigma$  will increase the number of tokens in  $\gamma$ . Thus  $M_0$  cannot be recovered. Since there exists no  $M_0$  and  $\sigma$  such that  $M_0 \xrightarrow{\sigma} M_0$ ,  $N$  is not consistent. Figure 8 shows two examples of inconsistent CFIO nets.

**Result 6:** If all elementary circuits of a CFIO net  $N$  are marked and all places in them have only one input transition, then the consistency of  $N$  can be determined by examining the consistency of a net  $N^1$  which is derived from  $N$  as follows: Replace each elementary circuit  $\gamma$  by a transition  $t_\gamma$  such that any input (output) place of a

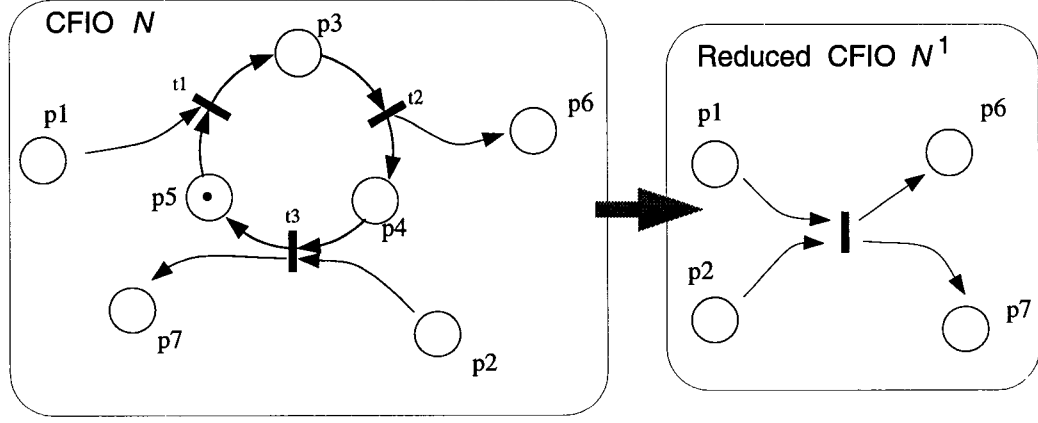


Figure 9: Circuit reduction rule

transition of  $\gamma$ , which is not included in  $\gamma$ , becomes an input (output) place of  $t_\gamma$ .  $N$  is consistent if and only if  $N_1$  is consistent.

**Proof of Result 6:** Let us assume that each elementary circuit contains at least one token, and that in any elementary circuit a place has only one input transition. We seek to prove that the reduction rule presented in Figure 9; i.e. the reduction of an elementary circuit to a single transition, as described in Result 6, preserves consistency. 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 consistency,  $\exists x \in \mathbb{N}^m$ , such that  $x > 0$  and  $Cx = 0$ .

Without loss of generality, we can assume that  $N$  contains only one elementary circuit  $\langle p_{i_1}, t_{j_1}, p_{i_2}, t_{j_2}, \dots, p_{i_s}, t_{j_s} \rangle$ . We assume that the rows (columns) of  $C$  are ordered according to the indices of the places (transitions). Then  $C^1$  is related to  $C$  as follows:

- Derive  $C^a$  from  $C$  by setting  $c_{i,j}^a = \begin{cases} c(i,j) & \text{for } j \neq j_1 \\ \sum_{k=1}^s c(i,j_k) & \text{for } j = j_1 \end{cases}$
- Derive  $C^b$  from  $C^a$  by removing the columns of transitions  $j_2, j_3, \dots, j_s$ .
- Obtain  $C^1$  by removing the rows of places  $i_1, i_2, \dots, i_s$  of  $C^b$ .

Since each  $p \in \gamma$  has one input and one output transition, each row  $i \in \{i_1, \dots, i_s\}$  in  $C$  is such that:

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

Furthermore  $Cx = 0$  implies:

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

Since this is true for all  $i \in \{i_1, \dots, i_s\}$ , it follows that  $x_{j_1} = x_{j_2} = \dots = x_{j_s}$ . Let us now build  $x^1$  as follows:

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

Based on the process followed to build  $C^1$ , the definition of  $x^1$ , and that fact that  $Cx = 0$ , we obtain  $C^1 x^1 = 0$ , which implies that  $N^1$  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.  $x_j = x_k^1$  if the  $k$ th column in  $C^1$  is the  $j$ th column in  $C$ .
- ii.  $x_{j_1} = x_{j_2} = \dots = x_{j_s} = x_k^1$  if transition  $t_k$  replaces the elementary circuit  $\gamma \subseteq N$  containing  $t_{j_1}, t_{j_2}, \dots, t_{j_s}$ .

Using the same arguments as in (a), we can see that  $x > 0$  and  $Cx = 0$ . This implies that  $N$  is consistent, which is a contradiction. Thus  $N_1$  is not consistent.

Q.E.D

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

### 3.3.2 Path Reduction

The second step in the CFIO reduction method is the *path reduction*, called as such because it detects and reduces paths that connect input to output transitions. Using path reduction all but the input and output transitions of the net are eliminated. Thus, in effect, the reduced net is a “black box” maintaining the input/output relation of the original net but, details on internal structure are abstracted. In most cases the resulting net is significantly reduced. The reduction maintains the consistency (or lack

of it) of the original net. Furthermore, the t-invariant information of the original net is retrievable from the reduced net. The small size of the final net facilitates the easy detection of consistency.

Since the path reduction algorithm operates on the output of the circuit reduction step, it is assumed that the CFIO net under consideration does not contain elementary circuits. The algorithm comprises the following two steps: i) define a set of relations between the number of firings of input and output transitions, and ii) construct the incidence matrix of the reduced net, based on these relations. In particular, each relation corresponds to one row of the matrix, i.e. one place of the reduced net.

### Path Reduction Procedure

**Step 1:** We define a set of relations between the number of firings of the input and output transitions. In the original net, for each transition  $t$ , we denote by  $n_t$  the number of times  $t$  can fire. If the initial marking of the CFIO net is zero, the following inequalities hold:

$$n_t \leq \sum_{u \in \bullet p} n_u, \quad \forall p \in \bullet t \quad (4)$$

If we apply inequality (4) repeatedly, starting from an output transition  $t_j$ , we obtain a set of inequalities between the numbers of input and output transition firings:

$$n_{t_j} \leq \sum_{k \in E} \beta_k^s n_{t_k} \quad \text{for } s = 1, 2, \dots, H_j \text{ and } j \in F \quad (5)$$

where:

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

Inequalities (5) provide the maximal number of times an output transition can fire, given the number of times the input transitions have been fired and assuming that the initial marking is zero.

**Step 2:** We will now show how to construct the incidence matrix of a reduced net such that:

1. The resulting net is consistent if and only if the initial net is consistent.
2. The resulting net contains only input and output transitions which have an one-to-one correspondence with the input and output transitions of the original net.
3. If the resulting net is consistent, each t-invariant of the resulting net corresponds a t-invariant of the original net.

Consider the consistency property of the net defined in Section 3.2.3. Since we assume zero initial marking, the CFIO net is consistent if and only if we can remove from it the tokens which have been introduced by firing the input transitions. Furthermore, since input transitions model the introduction of parts and raw materials to the system, we assume that they do fire at least once.

According to the process which leads to inequalities (5), 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 (5) turn into equalities, i.e.

$$-n_{t_j} + \sum_{k \in E} \beta_k^s n_{t_k} = 0 \quad (6)$$

Similarly, the CFIO net has a t-invariant if there exists  $n = [n_{k_1}, n_{k_2}, \dots, n_{k_L}]$ ,  $n \not\geq 0$ ,  $\{k_1, k_2, \dots, k_L\} = E \cup F$ , such that inequalities (5) turn into equalities. Note that if a Petri net is consistent, it has at least one t-invariant, but the converse is not true.

The incidence matrix  $C^1$  of the reduced net  $N^1$  is obtained starting from equations (6). Each equation in (6) provides one row of  $C^1$ . Consider the  $s^{\text{th}}$  equation related to  $t_j$  and the corresponding row of  $C^1$ :

- 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  $\beta_k^s$ ,
- the other elements of the row are 0.

The reduced CFIO net corresponding to the incidence matrix  $C^1$  is hereafter denoted by  $N^1$ , while the initial CFIO net is denoted by  $N$ . Similarly to  $N$ , the net  $N^1$  is consistent if inequalities (5) become equalities for strictly positive values. Furthermore  $N^1$  has a t-invariant if, similarly to  $N$ , inequalities (5) turn into equalities for values of  $n = [n_{k_1}, n_{k_2}, \dots, n_{k_L}]$ ,  $n \not\geq 0$ ,  $\{k_1, k_2, \dots, k_L\} = E \cup F$ .

The consistency of  $N$  is directly related to the consistency of  $N^1$ . Assume that  $x^1$  is known ( $x^1 \not\geq 0$ ) such that  $C^1 x^1 = 0$ , i.e.  $N^1$  is consistent. Firing the input transitions of  $N$  as many times as indicated in  $x^1$ , and then firing the enabled transitions until none remains enabled anymore, we obtain the following results :



- (i) the elements of  $x$  and  $x^1$  which correspond to the input and output transitions of  $N$  and  $N^1$ , respectively, are the same.
- (ii) the vector  $x$ , the elements of which represent the number of times each transition fired, satisfies  $Cx = 0$

The following result summarizes the above remarks:

**Result 7:** Given a CFIO net  $N$  with zero initial marking we define a relationship between the firings of its input and output transitions. In particular, this relationship provides the upper bound of the number of times each output transition fires as a positive linear combination of the number of times the input transitions fires. Considering these upper bounds, it is possible to derive a reduced net  $N^1$  as shown above. Furthermore:

- (i)  $N$  is consistent if and only if  $N^1$  is consistent.
- (ii) A vector  $x > 0$  such that  $Cx = 0$  is derived in a unique way from a vector  $x^1 > 0$  such that  $C^1x^1 = 0$ , and vice versa;  $C$  ( $C^1$ ) is the incidence matrix of  $N$  ( $N^1$ ).
- (iii) 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  $C^1x^1 = 0$ , and vice versa. Vector  $x$  ( $x^1$ ) is a t-invariant of  $N$  ( $N^1$ ).

In this subsection we have shown how the path reduction procedure can be used to further reduce a CFIO net that has already been reduced by the circuit reduction procedure. The resulting net is simpler, and it preserves the consistency property as well as the t-invariant information. An example illustrating the procedure follows:

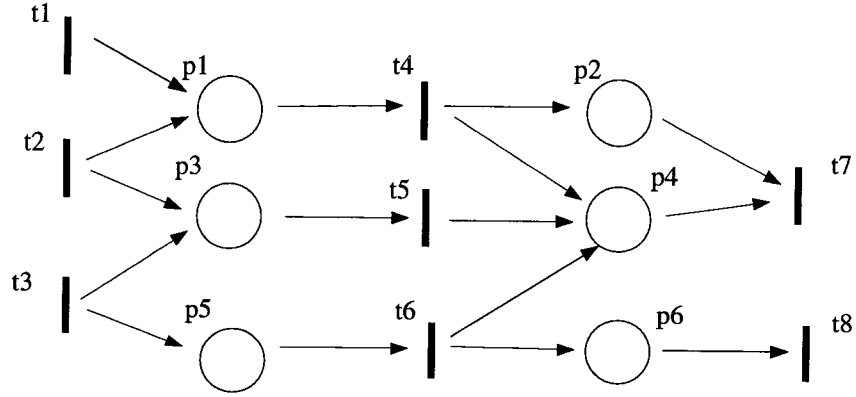


Figure 10: Initial net  $N$

**Example of Path Reduction Procedure:** Consider the CFIO net given in Figure

10. By applying inequality (4) to the output transition  $t_7$  we obtain:

$$n_7 \leq n_4 \quad (7)$$

$$n_7 \leq n_4 + n_5 + n_6 \quad (8)$$

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

$$n_4 \leq n_1 + n_2 \quad (9)$$

$$n_5 \leq n_2 + n_3 \quad (10)$$

$$n_6 \leq n_3 \quad (11)$$

Combining (7) with (9), and (8) with (9) through (11), we obtain, respectively:

$$n_7 \leq n_1 + n_2 \quad (12)$$

$$n_7 \leq n_1 + 2n_2 + 2n_3 \quad (13)$$

Similarly, starting from  $t_8$ , we obtain:

$$n_8 \leq n_3 \quad (14)$$

Inequalities (12) to (14) comply with inequalities (5).

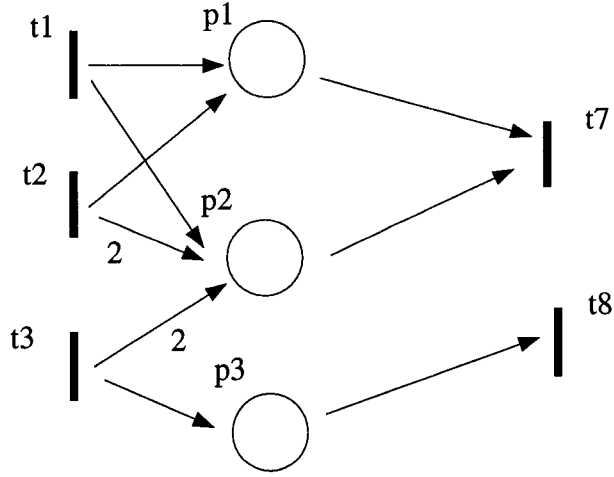


Figure 11: Reduced net  $N^1$

The related reduced net  $N^1$  is given in Figure 11.

In this example:

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

and

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

Let us examine the consistency of the reduced net and obtain the t-invariant information.  $N^1$  is not consistent, since there is no solution of  $C^1 x^1 = 0$  such that  $x^1 > 0$ . Thus, it follows that the original net  $N$  is not consistent. Note that vector  $x^1 = [1, 0, 0, 1, 0]$  is such that  $C^1 x^1 = 0$ . Thus,  $x^1$  is a t-invariant, which means that if we fire  $t_1$  once, starting from zero-marking, we return to zero marking by firing  $t_7$ . In this example inequalities (6) become equalities if  $n_1 = n_7 = 1$  and  $n_2 = n_3 = n_8 = 0$ . By firing  $t_1$  in  $N$ , we enable  $t_4$  and by firing  $t_4$  we enable  $t_7$ . Thus  $x = [1, 0, 0, 1, 0, 0, 1, 0]$  is a t-invariant of  $N$ .

### 3.4 Decomposition of a Petri Net Model of a Manufacturing System

CFIO nets can be used as building blocks for larger and more complicated nets that model realistic manufacturing systems. In this section we will show that the qualitative properties we discussed in the previous section propagate to Petri nets that are decomposable to CFIO nets.

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

- i)  $NC$  is a connected CFIO net.
- ii) The input (output) transitions of  $NC$  are input (output) transitions of  $N$ .
- iii) Every node (place or transition) of  $NC$  is a node of  $N$ .
- iv) For every transition  $t$  of  $NC$ , the set of input (output) places of  $t$  in  $NC$  is equal to the set of input (output) places of  $t$  in  $N$ .

**Definition 15:** We call a Petri net  $N$  with input and output transitions a *decomposable* Petri net, if there exist consistent CFIO nets  $NC_1, NC_2, \dots, NC_r$ , of  $N$ , such that:

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

**Result 8:** A decomposable Petri net each elementary circuit of which contains at least one token

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

**Proof:**

(i) Let  $C$  be the incidence matrix of the decomposable Petri net  $N$  and  $NC_1, NC_2, \dots, NC_r$  a set of consistent CFIO nets such that  $N = \bigcup_{i=1}^r NC_i$ . The consistency of  $NC_i$ ,  $i = 1, 2, \dots, r$  implies that there exists a vector  $x^i \in \mathbf{N}^m$  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  $x^* \in \mathbf{N}^m$  such that

$$x_k^* = \sum_{i=1}^r x_k^i \text{ for } k = 1, 2, \dots, m \quad (15)$$

then

$$Cx^* = C \sum_{i=1}^r x^i = \sum_{i=1}^r Cx^i = 0 \quad (16)$$

Furthermore

$$x_k^* = \sum_{i=1}^r x_k^i > 0 \text{ for } k = 1, 2, \dots, m \quad (17)$$

since every transition  $t_k$ ,  $k = 1, 2, \dots, m$  belongs at least to one  $NC_i$ , and, thus,  $x_k^i$  is strictly positive for at least one  $i \in \{1, 2, \dots, r\}$ .

From equations (16) and (17) we conclude that  $N$  is consistent.

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

$$x = \sum_{i=1}^r n_i x^i = [x_1, x_2, \dots, x_m] \quad (18)$$

where  $x^i$  are the vectors introduced in (i) above, and  $n_i$  are nonnegative integers. Then if transition  $t_k$  is fired  $x_k$  times,  $k = 1, 2, \dots, m$  (which is always possible) the resulting marking is the initial marking. Thus,  $N$  can be kept bounded if the transitions are fired according to (18).

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

Q.E.D.

Hereafter we will call *activation* of the CFIO net  $NC_i$  the firing of the transitions of  $NC_i$  as many times as required by the value of the corresponding component in  $x^i$

**Corollary 2:** From the proof of Result 8 we can deduce that by sequential activation of the manageable CFIO nets that compose the Petri net model we can ensure that the system will retain its expected qualitative properties as defined by manageability: liveness, boundedness and consistency.

### **3.5 Chapter Summary**

This chapter defines a subclass of black and white Petri nets, the Conflict Free nets with Input and Output transitions (CFIO). It also introduces the notion of manageability which is defined by a set of properties that are important in manufacturing systems i.e. absence of deadlocks and overflows, ability to reach any valid inventory level and any machine availability state, and ability to recover the initial system. Flexible manufacturing systems can be modeled by Petri nets that are unions of CFIO nets, and are manageable if the underlying CFIO nets can be kept bounded and are consistent. A reduction algorithm that simplifies the detection of these properties, given a union of CFIO nets, has been developed. In the following chapter we show how to use CFIO nets to evaluate the designs of a FMS. Furthermore, we employ CFIO nets to perform planning and scheduling.

## **4 Application of CFIO Nets to Flexible Manufacturing Systems**

In this chapter we first discuss the modeling of flexible manufacturing systems using CFIO nets. Subsequently, we use the Petri net model of a system to evaluate its design and to perform the production planning and scheduling tasks.

### **4.1 Petri Net Models of Flexible Manufacturing Systems**

We use a bottom-up synthesis approach for modeling flexible manufacturing systems. It involves determining the CFIO nets that compose a system, and, subsequently, synthesizing the Petri net model from these modules. There is no rigorous procedure to perform this bottom-up synthesis. Furthermore the Petri net model of a manufacturing system is not unique.

Our experience shows that a routing-based modeling approach is convenient in most cases. This approach considers the sequence of operations that are necessary to manufacture each part in the system (part routing). The operations included in a part routing, their inputs and outputs, and the related buffers, are modeled by a single CFIO net. The inputs and the outputs of this net correspond to the introduction of raw materials, and the delivery of the finished parts, respectively. Note that since we are considering flexible manufacturing systems, alternative routings for a part are possible. In this case, each alternative routing should be modeled by a separate CFIO net. The CFIO nets corresponding to the production routings of all parts are synthesized to form the system's Petri net model.



## Example

Consider an FMS that comprises three machines  $M_1, M_2, M_3$ , and manufactures two types of parts  $P_A$  and  $P_B$ . The manufacturing operation sequence for  $P_A$  is  $\langle O_1, O_2 \rangle$ , while the manufacturing operation sequence for  $P_B$  is  $\langle O_3, O_4 \rangle$ . Each manufacturing operation can be performed by one or more machines, i.e:

$$O_1 : \{M_1(4), M_2(1)\}, O_2 : \{M_3(2)\}, O_3 : \{M_1(4)\}, O_4 : \{M_2(2), M_3(4)\}$$

The quantities in parentheses are the related manufacturing times. Figure 12 shows the manufacturing operation sequences and the corresponding machines for both part types.

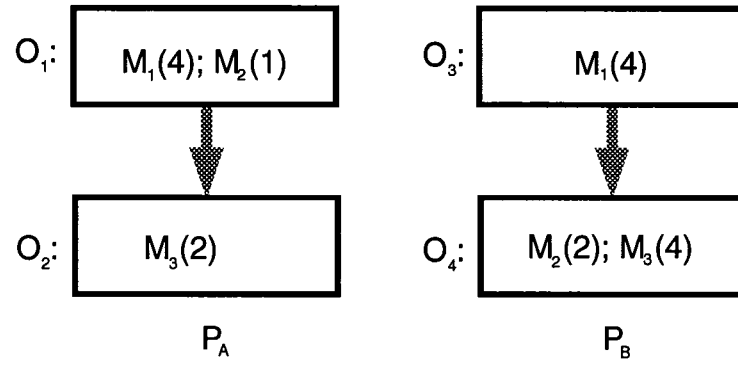


Figure 12: Manufacturing operation sequences for parts  $P_A$  and  $P_B$ .

Additional set-up operations are required in the manufacture of  $P_A$  and  $P_B$ . In operation  $O_3$  of product  $P_B$  an operator mounts a special fixture on machine  $M_1$  (operation  $S_1$ ). After operation  $O_3$  is completed, the operator removes the fixture from the machine (operation  $S_2$ ). If the operator becomes unavailable, a replacement operator is provided (operation  $S_3$ ) to perform the required tasks. Operation  $S_4$  represents the introduction of raw materials for part  $P_A$ ; no manufacturing time is associated with  $S_4$ .

Table 1 Transition to operation relation for CFIO nets in Figure 13

transition	operation	resource
$t_1$	$S_4$	—
$t_2$	$O_1$	$M_1$
$t_3$	$O_1$	$M_2$
$t_4$	$O_2$	$M_3$
$t_5$	$S_3$	operator
$t_6$	$S_1$	operator
$t_7$	$O_4$	$M_3$
$t_8$	$O_4$	$M_2$
$t_9$	$O_3$	$M_1$
$t_{10}$	$S_2$	operator

Table 2 Description of places for the CFIO nets in Figure 13

place	description
$p_1$	Part available for operation $O_1$
$p_2$	Part available for operation $O_2$
$p_3$	Operator available for operation $S_1$
$p_4$	Part available for operation $O_4$
$p_5$	Part and fixture available for $O_3$
$p_6$	Fixture available for operation $S_2$
$p_7$	Fixture available for operation $S_1$

The set-up related tasks are:

$S_1$ : Operator mounts fixture and product on  $M_1$ . The required set-up time is 2 time units.

$S_2$ : Operator removes fixture. The required tear down time is 1 time unit.

$S_3$ : Replacement operator becomes available (no time is required for  $S_3$ ).

$S_4$ : Raw material for part  $P_A$  becomes available (no time is required for  $S_4$ ).

The first step in modeling this system by CFIO nets is to identify the production routings. There are two alternative routings for part type  $P_A$ :  $R_A^1 = \langle M_1, M_3 \rangle$ ,  $R_A^2 = \langle M_2, M_3 \rangle$ . The routings for part type  $P_B$  are:  $R_B^1 = \langle M_1, M_2 \rangle$ ,  $R_B^2 =$

$\langle M_1, M_3 \rangle$ . We model each of these routings by a distinct CFIO net. This is done by:

- i) modeling each operation-resource combination by a transition (see Table 1), and ii)
- modeling each condition necessary for performing an operation by a place (see Table 2).

Figure 13 shows the CFIO nets corresponding to the two alternative routings of each part  $P_A$  and  $P_B$ . In this figure  $NC_i^j$  is the model of routing  $R_i^j$ .

The Petri net model of the manufacturing system includes the union of the CFIO

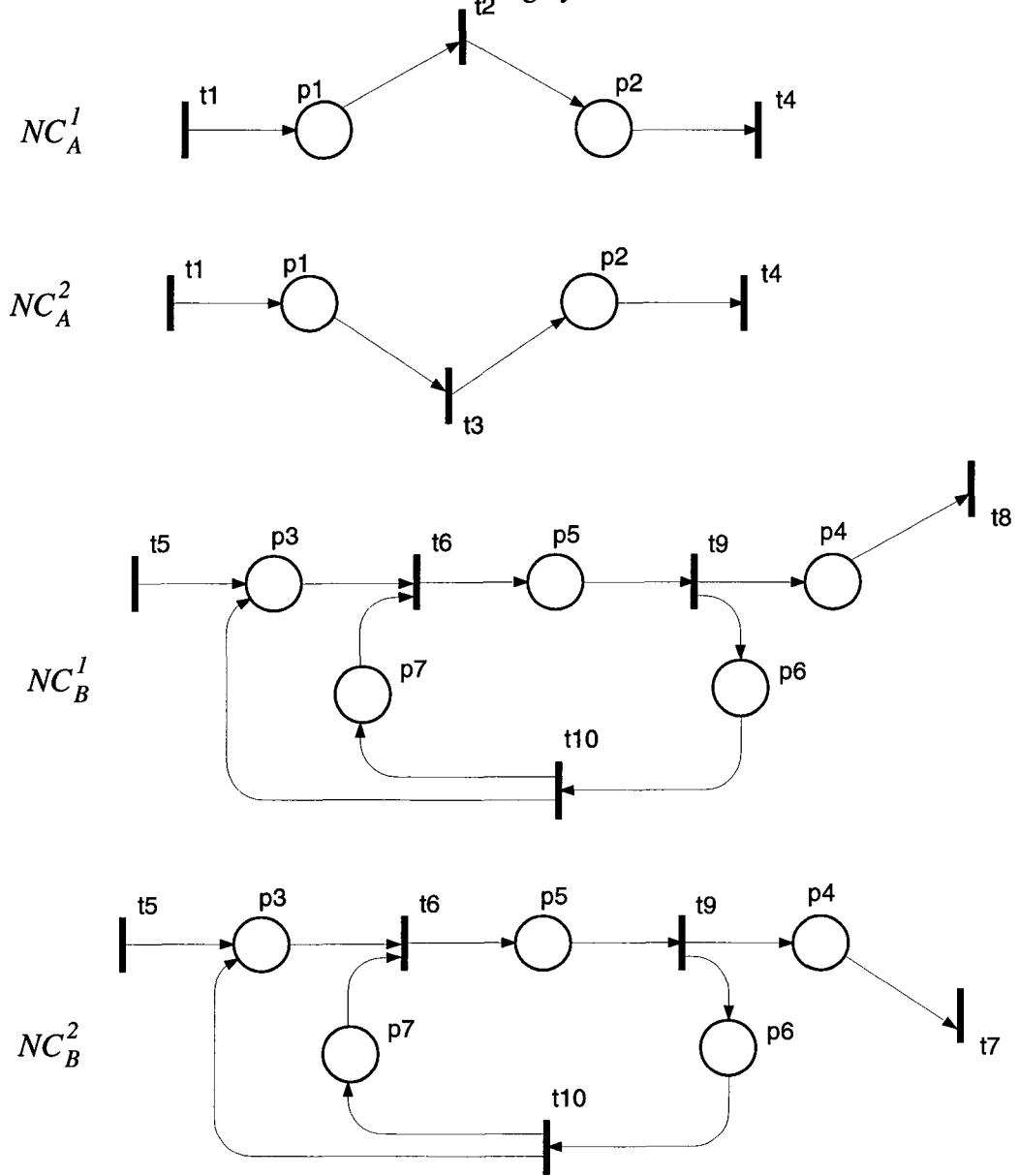


Figure 13: CFIO nets corresponding to the production routings of parts  $P_A$ ,  $P_B$ .

nets of Figure 13. We complete this model by introducing places  $q_1, \dots, q_6$  that relate the operations performed by the same machine. By placing only one token in each of these additional places we guarantee that the machines can perform at most one operation at any given time. Figure 14 shows the resulting model of the FMS.

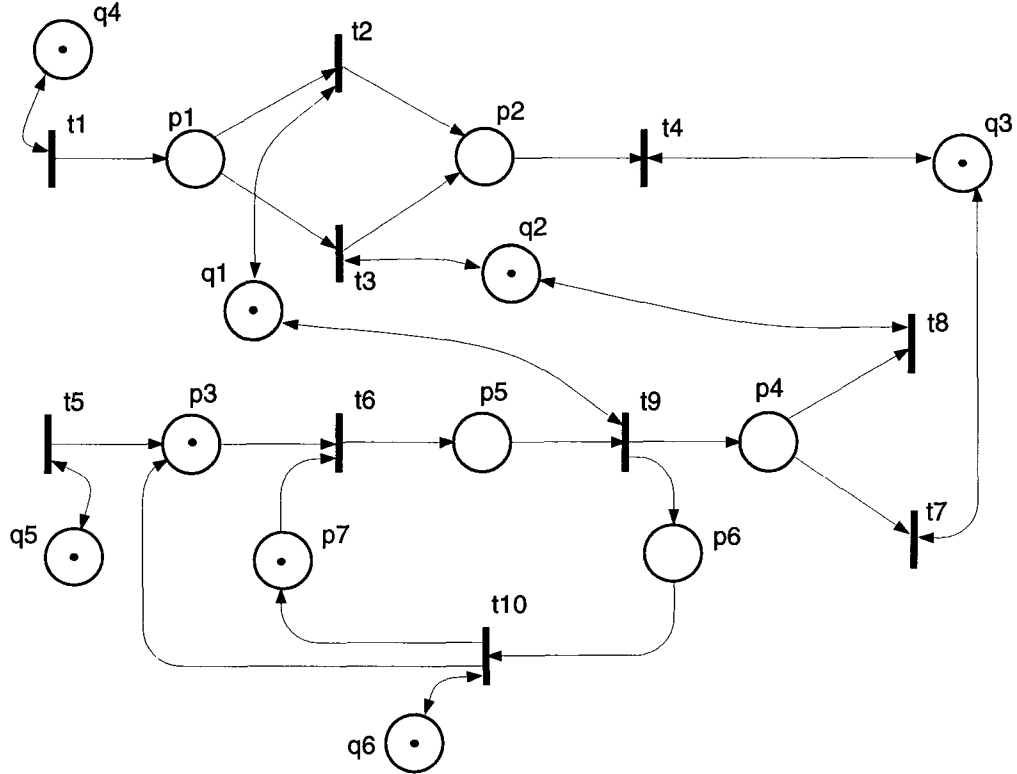


Figure 14: Petri net model of FMS including resource constraining places  $q_1, \dots, q_6$

## 4.2 Design Evaluation

The Petri net model of the manufacturing system is used to evaluate the system design with respect to manageability. This is accomplished by determining whether the CFIO nets of the model are consistent and can be kept bounded. As we have shown in the previous chapter, liveness is an intrinsic property of CFIO nets and guarantees the absence of deadlocks in the manufacturing system. Moreover, it ensures that the system can produce all parts for which it is designed for. Boundedness, on the other hand,

implies the absence of buffer overflows and guarantees that there are no uncontrollable subsystems present. Finally, consistency guarantees the ability to return to the initial state from any system state. This, in turn, implies that any valid inventory level, as well as any machine availability state, can be reached.

According to Chapter 3, determining the manageability of the Petri net model consists of the following two steps that have to be performed on all CFIO nets of the model:

1. **Circuit reduction:** If self-sustained circuits are detected, the net under consideration cannot be kept bounded and is, therefore, non-manageable. This implies that there is an uncontrollable subsystem in the manufacturing system, which is modeled by the self-sustained circuit. If no self-sustained circuits are found, then the system can be kept bounded by controlling its input transitions. During circuit-reduction, non-consistent circuits are also detected. If such circuits exist, the system is not consistent and, thus, non-manageable. This anomaly implies that there exists a buffer that may overflow. If none of the above anomalies are detected, we proceed with the path reduction step.
2. **Path reduction:** The net is reduced as described in Section 3.3.2. If the reduced net  $N^1$  is consistent, so is the initial net  $N$ , and, thus,  $N$  is manageable. Consistency of  $N^1$  (with incidence matrix  $C^1$ ) is determined by computing the extremal solutions of  $C^1 x^1 = 0$  using classical linear programming approaches ([4], [5]). This computation is usually very fast, due to the small size of  $C^1$ .

If one of the CFIO nets of the system model is found to be non-manageable,

the cause is detected. Subsequently, the designer has to determine if the problem is due to a modeling or a design error, and correct it accordingly. The procedure is repeated until all CFIO nets are determined to be manageable. This ensures that under sequential activation of the CFIO nets of the Petri net model the manufacturing system retains its qualitative properties as defined by manageability: liveness, boundedness, and reversibility (see Corollary 2).

### Example

The design evaluation approach described above is applied to the sample FMS introduced in Section 4.1.

**Circuit Reduction:** The circuit reduction method is applied to each of the CFIO nets of Figure 13.  $NC_A^1$  and  $NC_A^2$  have no circuits and remain unaltered. Consider the CFIO net  $NC_B^1$  and its circuit  $\langle t_6, p_5, t_9, p_6, t_{10}, p_3, t_6 \rangle$ . Note that place  $p_3$  of this circuit has more than one input transition, implying that the CFIO net is not consistent and that the corresponding system is not manageable. This anomaly forces us to search for the underlying system design flaws. Two such flaws are determined: i) There has been no consideration for removing the replacement operators once introduced to the system. ii) By modeling the introduction of an operator to the system by an input transition, we implicitly assume that there is an unlimited pool of replacement operators available. We will resolve these problems by establishing a pool of available operators (at least one), who are responsible for the fixture set-up and tear-down tasks required for operation  $O_3$ . Operators from this pool are assigned to these tasks as needed. When the tasks are complete the operators become part of the available pool.

The updated version of  $NC_B^1$  which reflects these design changes is shown in Figure 15. Place  $p_3$  models the pool of operators, and its tokens model the available operators. After operation  $S_2$  (modeled by  $t_{10}$ ) is completed, the operator rejoins the pool of available operators. Operation  $S_1$  (modeled by  $t_6$ ) is performed by an available operator from the pool.

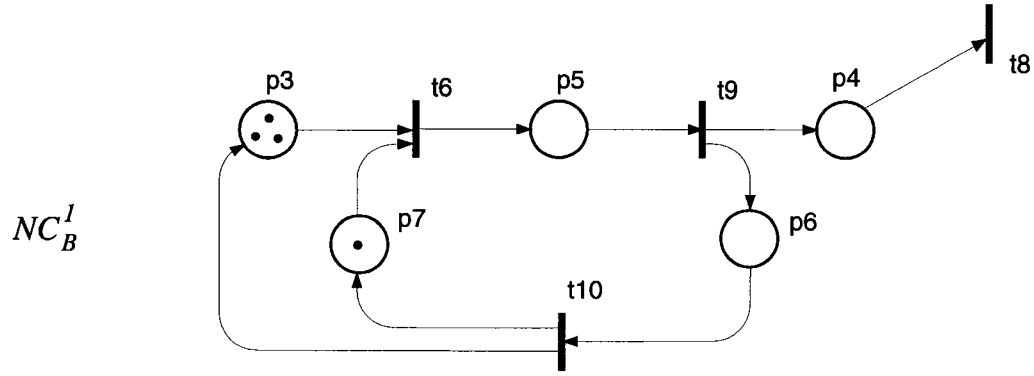


Figure 15: CFIO net  $NC_B^1$  modeling the redesigned routing  $R_B^1$

The CFIO net of Figure 15 has two circuits:  $\langle t_6, p_5, t_9, p_6, t_{10}, p_3, t_6 \rangle$  and  $\langle t_6, p_5, t_9, p_6, t_{10}, p_7, t_6 \rangle$ . These are reduced to a single transition in accordance to the circuit reduction rules, i.e. by substituting in the corresponding incidence matrix all the columns corresponding to the circuit transitions by a new column, that is equal to the sum of the ones removed. The reduced net has as follows:

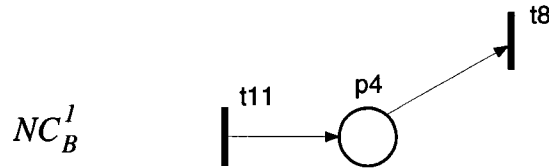


Figure 16: CFIO net  $NC_B^1$  after circuit reduction has been performed

The reduction resulted to an input transition,  $t_{11}$ , implying that the reduced circuit was a self-sustained circuit. This is due to a modeling error; i.e. we failed to model the introduction of raw materials for the corresponding routing; as a consequence the

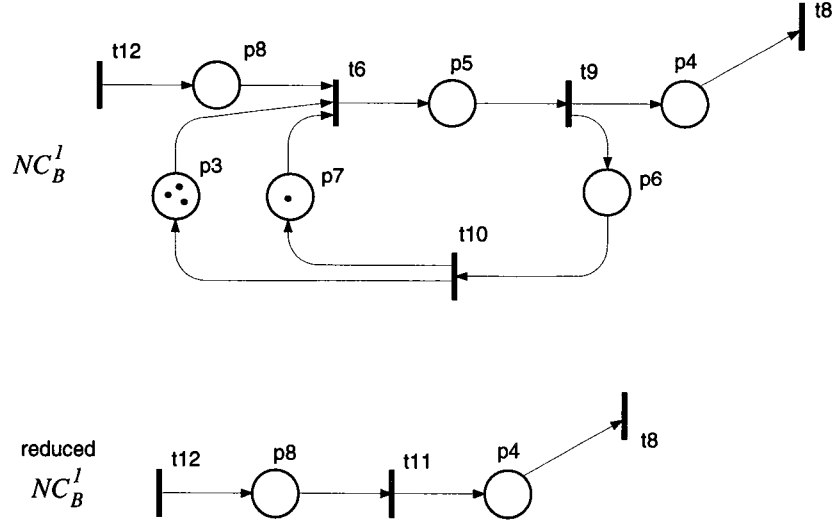


Figure 17: Final version of  $NC_B^1$

model implied that the underlying system could produce indefinitely. The corrected model and its reduced version are shown in Figure 17, where transition  $t_{12}$  models the introduction of raw materials denoted by a new operation  $S_5$  in the operation sequence of part  $P_B$ . As with operation  $S_4$ , we assume that there is no manufacturing time associated with  $S_5$ . Place  $p_8$  models the availability of raw materials. Transition  $t_{11}$  models the aggregated operations  $S_1, O_3$  and  $S_2$ . The firing time of  $t_{11}$  is equal to the total operating time for  $S_1, O_3$  and  $S_2$ , i.e. 4 time units. The resulting net has no other circuit, and the circuit reduction step is complete.

The circuit reduction of CFIO  $NC_B^2$  is similar to that of CFIO  $NC_B^1$ , and the corrected and reduced models of  $NC_B^2$  are shown in Figure 18.

**Path Reduction:** Consider the CFIO net  $NC_A^1$  in Figure 13. According to the path reduction procedure described in Section 3.3.2 we assume that the initial marking is zero and we define a set of inequalities between the number of firings of the input and output transitions (in this case between transition  $t_1$  and  $t_4$ ). To do so we apply



Equation (4) starting from the output transition,  $t_4$ , to obtain  $n_2 \leq n_4$  and  $n_2 \leq n_1$ .

The resulting input-output relation is

$$n_1 \leq n_4$$

which complies with Equation (5). The reduced net comprises a single place, one input and one output transition. This was expected, since CFIO  $NC_A^1$  was a single path connecting a single input transition to a single output transition. Obviously the reduced net is consistent, and thus CFIO net  $NC_A^1$  is also consistent. The path reduction procedure for the other three CFIO nets is similar and results to reduced nets, each comprising an input transition, a place and an output transition.

This analysis showed that after the design and modeling corrections all CFIO nets are manageable, and, so is the Petri net model of the manufacturing system (which is defined by their union). Figure 19 presents the final Petri net model of the manufacturing system, and incorporates the necessary corrections for the design flows and the modeling errors.

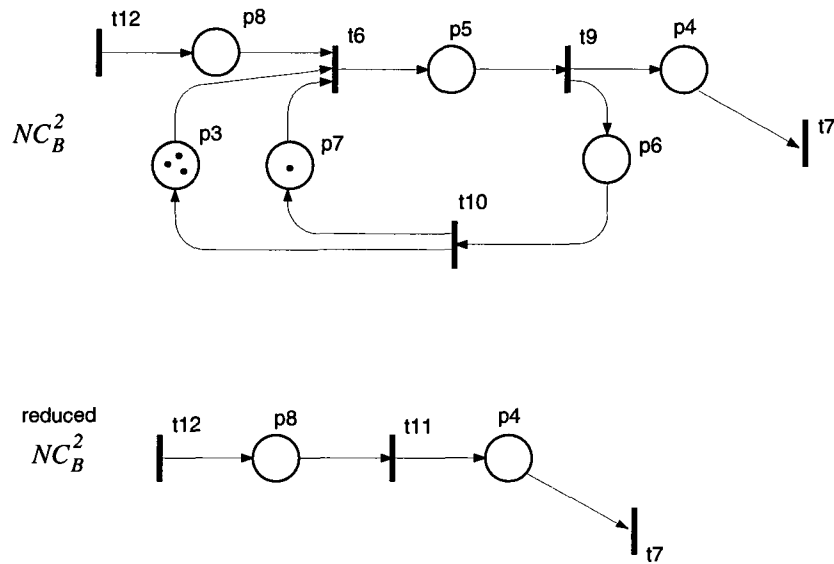


Figure 18: Final version of  $NC_B^2$

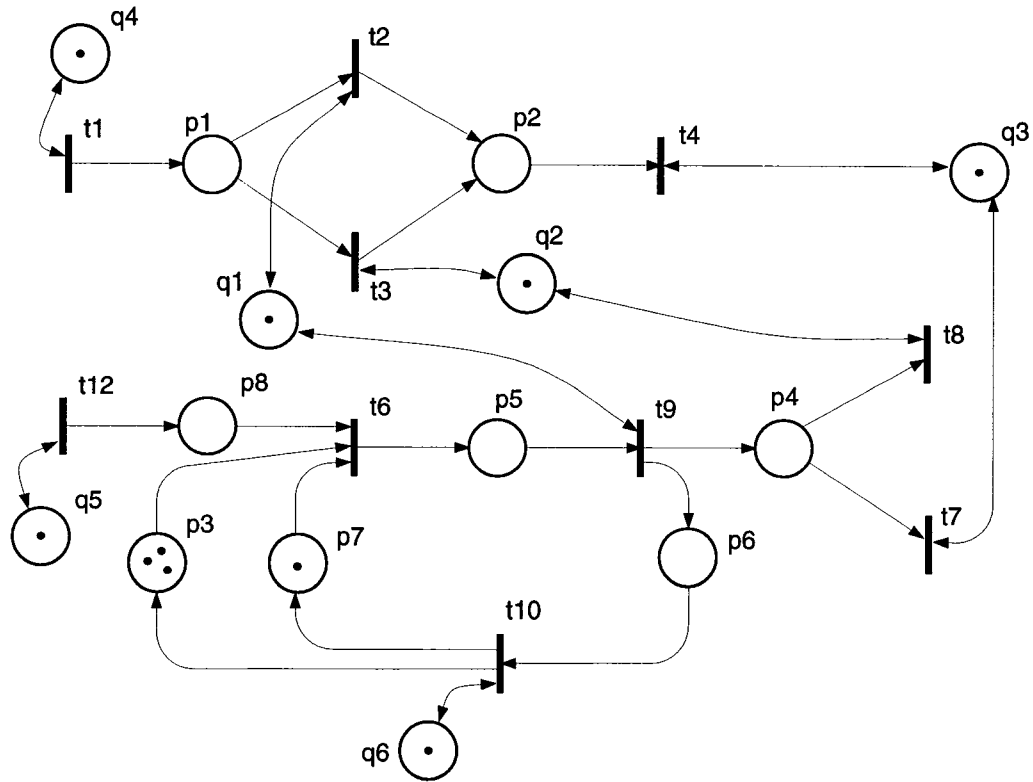


Figure 19: Model of manufacturing processes and resource constraints

### 4.3 Planning

Given a list of parts, the corresponding quantities required and the part due dates, production planning determines the time period in which each operation of each part should be performed, as well as the number of parts to be processed by each operation. Production is planned to optimize a system performance index, e.g. minimizing work-in-process inventory, balancing machine utilization, etc. In this section we show how to use the Petri net model of a flexible manufacturing system in order to develop a convenient linear programming formulation of the production planning problem. By solving the resulting linear program, we determine the number of times the CFIO nets have to be activated within each time period, so that the demand is satisfied and the criterion is optimized. Activation of a CFIO net corresponds to performance of the

operations of the corresponding routing during the activation period.

As will be shown in the next section, the advantage of using Petri nets (decomposable to CFIO nets) as the basis of the planning problem formulation is that it provides the means to abstract this problem in a convenient manner. Furthermore, it allows the planner to select the level of abstraction in order to obtain a desirable balance between solution quality and computational burden. Finally, using CFIO nets for planning guarantees the preservation of the qualitative properties of the system during system operation.

### 4.3.1 Planning Model

Our approach consists of two steps. First, given the Petri net model of the manufacturing system, we decompose it to CFIO nets. Secondly, we use the resulting CFIO nets to formulate the linear program of the planning problem, which is solved by standard techniques. Note that the linear program does not consider each operation of each part separately, but aggregates all operations corresponding to a single CFIO net into one decision variable. Correspondingly, the solution of the linear program provides the number of times each CFIO net should be activated within each time period, or, equivalently, the number of times the operations of the corresponding routings should be performed.

**Step 1:** A natural decomposition of the system's Petri net model to CFIO nets is achieved if each CFIO net corresponds to a routing of a single part. To reduce the computation burden of production planning, it may be desirable, however, to decompose the Petri net model, such that the individual CFIO nets represent more

than one routing. In this case, the number of the resulting CFIO nets is smaller and so is the size of the linear program. The drawback of considering unions of routings during production planning is that we reduce the flexibility of producing the corresponding parts in variable ratios. This is because by aggregating the CFIO nets of two or more parts we do not have the flexibility of activating their nets separately.

To better define this “production flexibility” issue we present the following result which is a direct consequence of Corollary 2 ( $N$  is the Petri net model of the manufacturing system under consideration):

**Result 9:** Let  $M \in R(M_0)$  such that  $M$  is reachable by sequential activation of the CFIO nets of  $N$ . Then it is possible to reach  $M_0$  from  $M$  by sequential activation of the same CFIO nets.

It is important to realize that the above result does not guarantee that  $M_0$  is reachable from all markings reachable by  $M_0$ , since not all  $M \in R(M_0)$  are reachable by sequential activation of the CFIO nets. The more CFIO nets we aggregate, the fewer markings are reachable by sequential activation of these nets. Thus, flexibility loss can be quantified by the number of markings which are reachable from the initial marking but are not reachable by sequential activation of the selected CFIO nets. Maximal flexibility would be obtained by using the minimal CFIO nets of the Petri net model, i.e., CFIO nets the transitions of which correspond to the *minimal* t-invariants. Minimal CFIO nets cannot be decomposed further. In fact, if the CFIO nets corresponding to  $N$  are the CFIO nets corresponding to the minimal t-invariants, then  $N$  is reversible for any marking  $M_0$  such that each elementary circuit contains at least one token.

Finally, it is emphasized that it is up to the designer to decide on the desired level of aggregation of the model's CFIO nets, in order to strike a good balance between computational burden and production flexibility.

**Step 2:** The second step of our production planning approach is to formulate the linear programming problem. Hereafter, we assume that the Petri net model of the manufacturing system is decomposable into manageable CFIO nets, and that such a decomposition has been established. As we stated above, production planning optimizes system performance while ensuring that a variable demand is satisfied. Therefore, our planning method should determine the number of times the CFIO nets of the system's model are activated, in order to satisfy the demand and optimize the planning criterion.

Activating a CFIO net involves firing transitions which correspond to operations of the flexible manufacturing system. Thus, each machine in the system should have adequate capacity to satisfy the workload requirements of these operations. The general formulation of the resulting capacity constraint for time period  $k$  is:

$$\sum_{j=1}^R \sum_{s \in O_{j,m}} z_s a_{j,k} \leq D_k \quad m = 1, \dots, H \quad k = 1, \dots, K \quad (19)$$

where

- $R$  is the number of CFIO nets,
- $O_{j,m}$  is the set of operations performed by machine  $m$  and modeled by a transition of CFIO  $j$ ,
- $z_s$  is the production time related to operation  $s$ ,
- $a_{j,k}$  is the number of times the  $j^{\text{th}}$  CFIO has to be activated during the  $k^{\text{th}}$  period,

- $D_k$  is the duration of the  $k^{\text{th}}$  time period,
- $H$  is the number of machines,
- $K$  is the number of time periods considered in the production planning problem.

The total demand for each part has to be satisfied by the cumulative production over all time periods. The resulting demand constraint has as follows:

$$\sum_{k=1}^K \sum_{j=1}^R u_{i,j} a_{j,k} \geq \sum_{k=1}^K d_k^i \quad i = 1, \dots, L \quad (20)$$

where

- $L$  is the number of parts
- $u_{i,j}$  is the number of parts  $i$  produced by routings modeled by CFIO  $j$
- $d_k^i$  is the demand for part  $i$  during period  $k$ .

In this work the optimization criterion consists of minimizing the sum of the inventory and backlogging costs. The general formulation is:

$$\mathcal{F} = \sum_{r=1}^K \sum_{i=1}^L \left\{ I_i \left[ \sum_{k=1}^r (-d_k^i + \sum_{j=1}^R u_{i,j} a_{j,k}) \right]^+ + B_i \left[ \sum_{k=1}^r (d_k^i - \sum_{j=1}^R u_{i,j} a_{j,k}) \right]^+ \right\}$$

where

- $I_i$  is the inventory cost for part  $i$ ,
- $B_i$  is the backlogging cost for part  $i$ ,
- $[x]^+$  is defined as  $\max(0, x)$

Let  $y_r^i$  be the cumulative inventory for part type  $i$  at the end of time period  $r$ , and  $z_r^i$  be the cumulative backlog of part  $i$  at the end of period  $r$ . Then

$$\begin{aligned} y_r^i &\geq \sum_{k=1}^r (-d_k^i + \sum_{j=1}^R u_{i,j} a_{j,k}) \\ z_r^i &\geq \sum_{k=1}^r (d_k^i - \sum_{j=1}^R u_{i,j} a_{j,k}) \end{aligned} \quad r = 1, \dots, K \quad i = 1, \dots, L \quad (21)$$

Using this notation the problem to solve becomes:

$$\text{Minimize : } \sum_{r=1}^K \sum_{i=1}^L (I_i y_r^i + B_i z_r^i)$$

subject to (19), (20), (21) and

$$a_{j,k} \geq 0 \quad j = 1, \dots, R; \quad k = 1, \dots, K$$

$$y_r^i \geq 0, \quad z_r^i \geq 0 \quad r = 1, \dots, K; \quad i = 1, \dots, L$$

The solution to the linear program provides the number of times each CFIO net needs to be activated in each planning period to minimize the inventory and backlogging cost and to satisfy the demand. In order to determine from this solution the production plan, i.e. the number of times each operation needs to be performed within each time period, we need to derive the number of times each transition fires during each time period. Since every transition models an operation-machine combination, the number of times each transition fires provides how many times each operation should be performed, and by which machine. The relationship between CFIO net activation and transition firing is given by the t-invariant of the corresponding CFIO net. In particular, the number of times a transition fires is given by the product of the number of times the CFIO net is activated and the value of the t-invariant element that corresponds to this transition. Since the CFIO nets are consistent, it is guaranteed that at least one t-invariant exists for each CFIO net. The CFIO net activations are provided by the solution of the linear

program and the t-invariants are determined from the reduced CFIO net as described in Result 7.

By performing these calculations we generate the production plan. Note that the order of the transition firings in each time period of the plan has not yet been determined and it may depend on other constraints (such as resource constraints). This ordering is determined by the scheduling activity discussed in Section 4.4.

The following example demonstrates our production planning approach.

### 4.3.2 Example

Consider the manufacturing system introduced in Section 4.1, and its Petri net model. The demand of parts  $P_A$  and  $P_B$  is given in Table 3 for four consecutive time periods. The part inventory and backlogging costs are given in Table 4. We assume that the length of each time period is 40 time units.

Table 3: Demand of parts  $P_A$  and  $P_B$

Period	Product Demand	
	$d_k^1$	$d_k^2$
$k = 1$	2	6
$k = 2$	15	1
$k = 3$	7	4
$k = 4$	3	17

Table 4: Costs for parts  $P_A$  and  $P_B$

	$P_A$	$P_B$
Backlogging Cost: $B_i$	20	30
Inventory Cost: $I_i$	4	2

In this example we will consider the Petri net model decomposition into the four CFIO nets as described in the previous section. Taking into consideration the manufacturing times for each operation-resource combination and the CFIO nets of the model, the capacity constraints for each machine  $M_1, M_2$  and  $M_3$  are derived from



Equation (19):

$$\begin{aligned}
4a_{1,k} &+ 4a_{3,k} + 4a_{4,k} \leq 40 \\
a_{2,k} &+ 2a_{3,k} \leq 40 \\
2a_{1,k} &+ 2a_{2,k} + 4a_{4,k} \leq 40
\end{aligned} \quad k = 1, \dots, 4 \quad (22)$$

Taking in consideration the product demand for each planning period (Table 3), the demand constraints for each product are derived from Equation (20):

$$\begin{aligned}
\sum_{k=1}^4 (a_{1,k} + a_{2,k}) &\geq 27 \\
\sum_{k=1}^4 (a_{3,k} + a_{4,k}) &\geq 28
\end{aligned} \quad (23)$$

The criterion of minimizing the sum of the inventory and backlogging costs is derived from the general formulation.

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

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

By setting

$$\begin{aligned}
y_r^1 &\geq \sum_{k=1}^r (a_{1,k} + a_{2,k} - d_k^1) \\
z_r^1 &\geq \sum_{k=1}^r (d_k^1 - a_{1,k} - a_{2,k}) \\
y_r^2 &\geq \sum_{k=1}^r (a_{3,k} + a_{4,k} - d_k^2) \\
z_r^2 &\geq \sum_{k=1}^r (d_k^2 - a_{3,k} - a_{4,k})
\end{aligned} \quad r = 1, \dots, 4 \quad (24)$$

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 (22), (23), (24) and

$$a_{i,k} \geq 0 \quad i, k = 1, \dots, 4$$

$$y_r^s \geq 0, z_r^s \geq 0 \quad r = 1, \dots, 4; s = 1, 2$$

From the solution of the linear program we obtain the values for  $\alpha_{i,k}$  i.e. the number of times the  $i^{\text{th}}$  CFIO has to be activated during the  $k^{\text{th}}$  period (see Table 5). Note that activating a CFIO net corresponds to performing the operations of the corresponding routing.

Table 5: Linear program solution: The number of times each CFIO needs to be activated during each time period

Period	$NC_A^1$	$NC_A^2$	$NC_B^1$	$NC_B^2$
$k = 1$	2	0	6	0
$k = 2$	10	5	2	0
$k = 3$	7	0	10	0
$k = 4$	3	0	10	0

To determine the production plan, it is necessary to calculate how many times each transition fires. This will provide the number of times each operation has to be performed, and by which machine. To this end we perform the following calculations.

Consider the CFIO net  $NC_A^1$  given in Figure 13. In Section 4.2 we have shown that by applying the circuit and path reduction methods, the CFIO net is reduced to a single place with one input and one output transition. According to result 7, these transitions correspond to the input and output transitions of CFIO net  $NC_A^1$ . The incidence matrix of the reduced net is therefore:

$$C^1 = p_8 \begin{bmatrix} t_1 & t_4 \\ 1 & -1 \end{bmatrix}$$

It is easy to observe that the vector

$$x^1 = \begin{matrix} < t_1, & t_4 > \\ < 1, & 1 > \end{matrix}$$

is such that  $C^1 x^1 = 0$ , i.e.  $x^1$  is a t-invariant for the reduced net. In accordance to Result 7, a vector  $x > 0$  such that  $Cx = 0$  always exists and can be derived from vector  $x^1$ , where  $C$  is the incidence matrix of the CFIO net  $NC_A^1$ . This is done by firing the input transition  $t_1$  of  $NC_A^1$  as many times as specified by  $x^1$ , i.e. once, and then fire all enabled transitions. The resulting t-invariant is:

$$\begin{matrix} < t_1, & t_2, & t_4 > \\ < 1, & 1, & 1 > \end{matrix}$$

By padding with zeros for the transitions not in  $NC_A^1$  we obtain

$$H_A^1 = \begin{matrix} < t_1, & t_2, & t_3, & t_4, & t_6, & t_9, & t_{10}, & t_{12}, & t_7, & t_8 > \\ < 1, & 1, & 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0 > \end{matrix}$$

Similarly, we obtain the corresponding t-invariant vectors for the other CFIO nets of this example:

$$\begin{matrix} < t_1, & t_2, & t_3, & t_4, & t_6, & t_9, & t_{10}, & t_{12}, & t_7, & t_8 > \\ H_A^1 = & < 1, & 1, & 0, & 1, & 0, & 0, & 0, & 0, & 0 > \\ H_A^2 = & < 1, & 0, & 1, & 1, & 0, & 0, & 0, & 0, & 0 > \\ H_B^1 = & < 0, & 0, & 0, & 0, & 1, & 1, & 1, & 1, & 0, & 1 > \\ H_B^2 = & < 0, & 0, & 0, & 0, & 1, & 1, & 1, & 1, & 1, & 0 > \end{matrix}$$

The number of times a transition fires is given by the product of the number of times the CFIO net is activated and the value of the t-invariant element that corresponds to this transition. From the above t-invariant information and Table 5 we obtain the number of times each transition has to be fired during each elementary period (see Table 6).

Based on Table 6, and considering Table 1 (which provides the relation between each transition and each operation-machine combination) we can derive the number of

Table 6: Planning Results: transition firings for each period

	Transition									
Period	$t_1$	$t_2$	$t_3$	$t_4$	$t_6$	$t_9$	$t_{10}$	$t_{12}$	$t_7$	$t_8$
$k = 1$	2	2	0	2	6	6	6	6	0	6
$k = 2$	15	10	5	15	2	2	2	2	0	2
$k = 3$	7	7	0	7	10	10	10	10	0	10
$k = 4$	3	3	0	3	10	10	10	10	0	10

Table 7: Production plan

	Operations and Machines									
Period	$S_4$	$O_1$ by $M_1$	$O_1$ by $M_2$	$O_2$ by $M_3$	$S_1$ by opera- tor	$O_3$ by $M_1$	$S_2$ by opera- tor	$S_5$	$O_4$ by $M_3$	$O_4$ by $M_2$
$k = 1$	2	2	0	2	6	6	6	6	0	6
$k = 2$	15	10	5	15	2	2	2	2	0	2
$k = 3$	7	7	0	7	10	10	10	10	0	10
$k = 4$	3	3	0	3	10	10	10	10	0	10

times each operation has to be performed by a certain machine per time period, i.e. the production plan (see Table 7).

The production during each period (Table 8) is derived from Table 7 if we consider that operation  $O_2$  produces part  $P_A$  and operation  $O_4$  produces part  $P_B$ . By comparing Tables 3 and 8, one can observe that the production of  $P_A$  followed the demand of  $P_A$  exactly. However,  $P_B$  was overproduced during periods  $k = 2$  and  $k = 3$  to satisfy the high demand during period  $k = 4$ . The incurred inventory costs were necessary, since the demand during period  $k = 4$  could not be satisfied by the production during that period due to resource capacity constraints.

Table 8: Production quantities

Period	Product	
	$P_A$	$P_B$
$k = 1$	2	6
$k = 2$	15	2
$k = 3$	7	10
$k = 4$	3	10

#### 4.4 Scheduling

In this section we present an approach to solve the scheduling problem, given a manageable FMS, its Petri net model and the production plan. The schedule defines the starting time of each operation within the time period the operation is to be performed as determined by the production plan.

The problem under consideration is scheduling of a *flexible job-shop*. *Job-shop* refers to the fact that different products follow different sequences of operations through the shop; *flexible* refers to the fact that more than one machine is capable of performing an operation. According to the standard notation of Blazewicz *et al.* [50] this problem can be denoted as  $J/res \bullet 11/C_{max}$ . It should also be noted that we deal with variable demand and non-zero WIP inventory.

There are two types of decisions that have to be made in order to solve the scheduling problem: (i) *Resource Use (RU)* decisions which consist of selecting the resource to perform a given operation when several resources are available. (ii) *Product Sequencing (PS)* decisions, which are concerned with the sequencing of product operations on each resource. The problem objective is to determine the *RU* and *PS* decisions within each time period of the production plan that result in a production

makespan that is *less* than the duration of the corresponding period. By production makespan we mean the time elapsed from the start of the first operation of the schedule to the time the last operation is completed. Note that in this work we do not seek to find the optimum schedule. Instead, we are trying to find a schedule that fits within the time periods of the production plan. To this end we propose an algorithm that tries to minimize the makespan; however any solution with a makespan lower than the duration of the time period is acceptable.

#### 4.4.1 Petri Net Models for Scheduling

Scheduling is based on the same Petri net models we developed for design evaluation and production planning. Some aspects of these models that are important for scheduling are discussed below; in particular we focus on the modeling of *RU* and *PS* decisions.

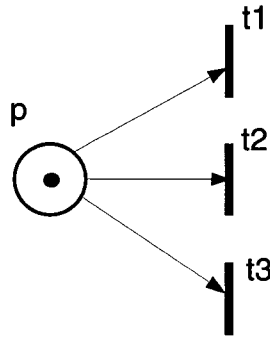


Figure 20: Modeling of RU type decisions

***RU* decisions** are modeled by places which have more than one output transitions (hereafter called *RU* places). Consider, for example, Figure 20: A decision has to be made whether the next operation for the part (modeled by the token in place *p*) has to be performed using the resource related to  $t_1$ , or the resource related to  $t_2$ , or the resource related to  $t_3$ . It is clear from our previous discussion on modeling of manufacturing

systems that a Petri net model which includes such a place will be decomposed to at least three CFIO nets, each containing the place  $p$  and one of its output transitions. The planning method determines how many times each of these three CFIO nets has to be activated within each time period. It is the task of scheduling to determine *when* each of those activations will occur within each period.

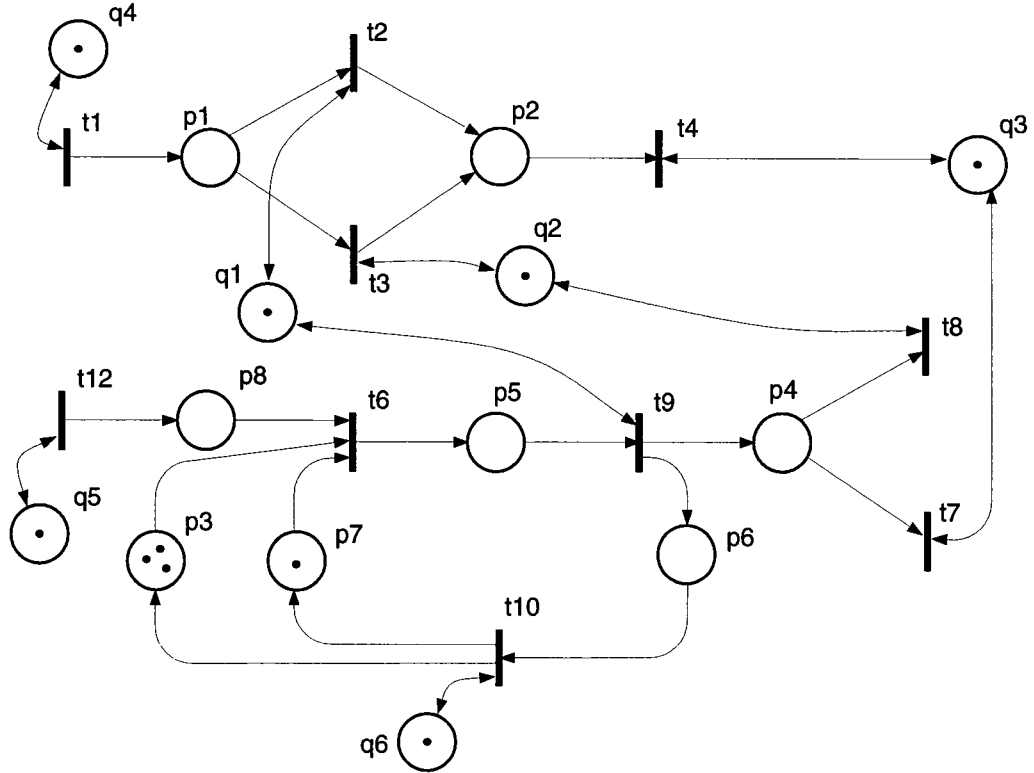


Figure 19: Model of manufacturing processes and resource constraints

In the Petri net model of the FMS shown in Figure 19 (and repeated here) there are two *RU* places,  $p_1$  and  $p_4$ . Place  $p_1$  and its output transition  $t_2$  belong to CFIO net  $NC_A^1$ . Place  $p_1$  and its output transition  $t_3$  belong to CFIO net  $NC_A^2$ . According to the result of production planning the number of times  $t_2$  and  $t_3$  are fired per period is known. For example in the second time period,  $t_2$  and  $t_3$  fire 10 and 5 times, respectively (see Table 6). In this section we will show how to determine when those activations should

occur, in order to comply with the production plan of each time period.

**PS decisions** are modeled by places that i) have more than one output, and ii) each output transition of such a place is also an input transition of the place (these places are hereafter called *PS places*). In Section 4.1 we introduced *PS* places to guarantee that each machine can perform at most one operation at any given time. In this Section we focus on the decision involved with these places. Consider, for example, Figure 21. A decision has to be made whether the resource (the availability of which is represented by the token in  $p$ ) should be used to perform the operation related to  $t_1$  or the operation related to  $t_2$ .

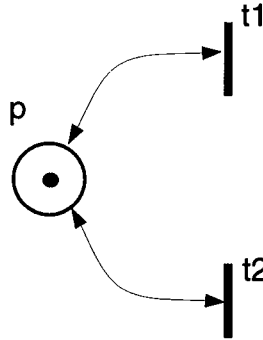


Figure 21: Modeling of PS type decisions

We represent a scheduling decision related to an *RU* or *PS* place by a sequence of transitions to be fired during the time period under consideration. All transitions of this sequence are output transitions of the *RU* or *PS* place. To respect the results of production planning, transitions appear in the sequence as many times as specified by the production plan. To illustrate the generation of a transition sequence consider place  $q_2$  of Figure 19 which is a *PS* place. It models the availability of machine  $M_2$ , which performs operation  $O_1$  (modeled by transition  $t_3$ ), and operation  $O_4$  (modeled by transition  $t_8$ ). According to the production plan, in the second time period,  $t_3$  and



$t_8$  fire 5 and 2 times respectively (see Table 6). Thus, for place  $q_2$  a valid scheduling sequence for this period is  $\sigma_{q_2} = \langle t_3, t_8, t_3, t_3, t_3, t_8, t_3 \rangle$ .

Using this notation the scheduling decisions are expressed in terms of sequences  $\sigma_p$  assigned to each *RU* and *PS* place of the net. Each sequence  $\sigma_p$  provides the order that the output transitions of  $p$  should be fired. If  $p$  is a *PS* place,  $\sigma_p$  indicates the order the output transitions of  $p$  should fire. Note that a given transition in the sequence will not start firing until the previous one stops firing. This is because each resource can process only one part at a time. If  $p$  is an *RU* place,  $\sigma_p$  indicates the order the output transitions of  $p$  should *start* firing. Note that a transition may start firing as soon the previous one has fired, since resources are allowed to operate concurrently. In other words, there may be an overlap in the firing period of transitions of  $\sigma_p$  if  $p$  is an *RU*-place, but no overlap is allowed if  $p$  is a *PS* place.

We assume that transitions are fired as soon as they are enabled. Consequently, a schedule is defined as soon as a sequence  $\sigma_p$  of transitions is assigned to each *RU* and each *PS* place.

It is important to note that the Petri nets we use for modeling flexible manufacturing systems are non-deterministic. However, resource usage and product sequencing decisions remove the ambiguity from the model, so that when the Petri net and the results of scheduling are considered together they result in a deterministic model.

#### 4.4.2 Scheduling Algorithm

Having established the appropriate system model, the objective of the scheduling problem can be expressed as follows: Determine the sequences  $\sigma_p$  to be assigned to

the *RU* and *PS* places that result in a makespan which is less than the duration of the planning period. Note that each planning period may have different duration and will be considered separately.

Since the scheduling problem of the general job-shop is NP-hard [18], only heuristic algorithms can be considered for solving large-sized problems. In this work we follow a critical path-based scheduling approach. Note that the critical path comprises the transitions that define the production makespan. Thus, in order to shorten the makespan, transitions belonging to the critical path should be adjusted.

The scheduling algorithm presented in this section will be employed repeatedly to determine the production schedule for each period of the production plan. For a given planning period we employ the following notations:

- $D$ : Period duration, in time units.
- $n_{t_i}$ : The number of firings of transition  $t_i$  within the planning period under consideration. If  $t_i$  models operation  $O_u$  on machine  $M_v$ ,  $n_{t_i}$  is the number of times operation  $O_u$  is performed on machine  $M_v$  within the planning period.
- The pair  $(t_i, k)$  denotes the  $k^{\text{th}}$  firing of transition  $t_i$ . If  $t_i$  models operation  $O_u$  on machine  $M_v$ ,  $(t_i, k)$  models the  $k^{\text{th}}$  time machine  $M_v$  performs operation  $O_u$ . Hereafter, the pair  $(t_i, k)$  will be referred to as a transition firing.
- $S_{t_i}(k)$  : Start time of  $(t_i, k)$ .
- $F_{t_i}(k)$  : Finish time of  $(t_i, k)$ :  $F_{t_i}(k) = S_{t_i}(k) + Z(t_i)$ , where  $Z(t_i)$  is the firing time of transition  $t_i$ .

The above notation is used in the following definitions.

**Definition 16:** A critical path is a sequence of transition firings

$$CP = \langle (t_{\alpha_1}, k_{\beta_1}), (t_{\alpha_2}, k_{\beta_2}), \dots, (t_{\alpha_n}, k_{\beta_n}) \rangle$$

such that:

1.  $S_{t_{\alpha_1}}(k_{\beta_1}) = 0$ .
2.  $F_{t_{\alpha_i}}(k_{\beta_i}) = S_{t_{\alpha_{i+1}}}(k_{\beta_{i+1}})$  ,  $i = 1, \dots, n - 1$
3.  $F_{t_{\alpha_n}}(k_{\beta_n}) = \max_{t,k} \{F_t(k)\}$ . The value of  $F_{t_{\alpha_n}}(k_{\beta_n})$  is the *makespan*.

Given the marked Petri net model and the starting time of every transition firing, the critical path can be easily computed starting from transition firing  $(t_{\alpha_n}, k_{\beta_n})$  which is the last firing in the path. The next to last transition firing in the path is the one with finishing firing time equal to  $S_{t_{\alpha_n}}(k_{\beta_n})$ . The computation continues in a backward fashion to include transition firings in the critical path until a firing with zero starting firing time is found. Note that the critical path may not be unique.

**Definition 17:** Let transition  $t_i$  model operation  $O_u$  on machine  $M_v$ . Then *float time*  $\mathcal{F}_{t_i}(k)$  of the transition firing  $(t_i, k)$  is the maximal time the transition firing can be delayed without increasing the makespan; i.e.

$$\mathcal{F}_{t_i}(k) = \min \left[ \left( \mathcal{F}_{t_j}(l) + S_{t_j}(l) - F_{t_i}(k) \right), \left( \mathcal{F}_{t_h}(m) + S_{t_h}(m) - S_{t_i}(k) \right) \right]$$

where

- transition firing  $(t_j, l)$  models the next scheduled operation by machine  $M_v$ , and
- transition firing  $(t_h, m)$  models the next scheduled machine to perform operation  $O_u$  (in an FMS it is possible that more than one machine can perform a specific operation).

This recursive definition ensures that the float time associated with the  $k^{\text{th}}$  time machine  $M_v$  performs operation  $O_u$  (denoted by  $(t_i, k)$ ) cannot be greater than: i) the float time of the next operation on  $M_u$  plus the difference between the ending time of  $(t_i, k)$  and the starting time of that next operation, or ii) the float time of the next machine scheduled to perform operation  $O_u$ , plus the difference between the starting times of  $(t_i, k)$  and the starting time of the next machine scheduled to perform operation  $O_u$ .

Float times can be calculated in a recursive manner, starting from an input transition. Consider the transition firing  $(t_i, k)$  and the *PS* and *RU* sequences that contain  $t_i$ . Given this information, recursively compute i) the float times of transition firing  $(t_j, l)$ , which is the next element in the *PS* sequence, and ii) the float time of transition firing  $(t_h, m)$  which is the next element in the corresponding *RU* sequence. This calculation of the float time takes into account the schedule and machine precedence constraints of the operations. Note that all transition firings on the critical path have zero float time.

The scheduling approach of this work assumes that the following information is available:

1. The Petri net model  $N$  of the manufacturing system, such that  $N$  is decomposable to manageable CFIO nets.
2. An initial marking,  $M_0$ , such that each elementary circuit contains at least one token.  $M_0$  models the initial WIP of the system.
3. Planning information including the duration  $D$  of the planning period under consideration, and the number of times,  $n_t$ , each transition,  $t$ , fires within this

planning period  $D$ .

The scheduling algorithm starts with a feasible set of sequences  $\sigma_p$ , for each  $PS$  and  $RU$  place, i.e. with a set of  $\sigma_p$ , which do not lead to blocking. To generate such a set we arbitrarily select an enabled transition  $t_i$ , and fire it (if it has not fired  $n_t$  times already). Since the net is *manageable*, it is guaranteed that this procedure will not deadlock and all transitions may fire as many times as specified by the planning results. The resulting sequence of firings determines the sequences  $\sigma_p$ , for all  $RU$  and  $PS$  places. To reduce the computation burden we assume that after a number of recursions the float times are zero.

The second step is to determine the *critical path* for this set of sequences  $\sigma_p$ , as discussed above. If the makespan of the critical path is greater than the duration of the planning period, it is evident from definition 16 that is *necessary* (but not sufficient) to reduce  $F_{t_{\alpha_n}}(k_{\beta_n})$ . Equivalently, it is necessary to move earlier in time the finishing time of one of the transition firings that belongs to the critical path. This, however, implies that another transition firing will be delayed. If the delay time is less than or equal to the float time, the transition can be delayed without increasing the makespan. Note that in doing so it is not allowed to violate the machine precedence constraints (i.e. the constraints related to the manufacturing process, taking into account the initial marking). The following algorithm is derived from the above remarks.

**Algorithm:** Consider the marked Petri net model of the manufacturing system, the solution to the production planning problem, and the duration of the planning period under consideration.

1. Compute a feasible set of sequences  $\sigma_p$ , and the starting times of the transition firings (as described above).
2. Compute the critical path and the makespan.
3. Select  $(t_u, k_u)$  and  $(t_c, k_c)$  such that:
  - a.  $t_u \neq t_c$ ;
  - b.  $(t_u, k_u)$  belongs to the critical path;
  - c.  $(t_c, k_c)$  immediately precedes  $(t_u, k_u)$  in one of the sequences  $\sigma_p$ , belonging to the feasible set of sequences;
  - d. swapping  $(t_u, k_u)$  and  $(t_c, k_c)$  in the sequence does not violate the machine precedence constraints;
  - e. the float time associated with  $(t_c, k_c)$  is greater than or equal to the delay time of  $(t_c, k_c)$  if the latter is swapped with  $(t_u, k_u)$  in the sequence  $\sigma_p$ .

If several pairs  $(t_u, k_u)$  and  $(t_c, k_c)$  satisfy the above conditions, select the one for which the associated delay is the closest to the float time.

4. If a pair has been found, swap  $(t_u, k_u)$  with  $(t_c, k_c)$  in the sequence  $\sigma_p$ , and go to step 2; else terminate the algorithm.

The output of the algorithm is a transition firing sequence for each *RU* and *PS* place. Those sequences, along with the Petri net model, uniquely define the starting time of each transition firing, i.e. the time each operation has to be performed on a machine.

Note that the final makespan may be greater than the duration of the planning period. If this is the case the time period needs to be increased, and the production plan and production schedule should be recalculated.

### 4.4.3 Scheduling Examples

The following examples demonstrate the above scheduling algorithm. The first example considers the manufacturing system we introduced in Section 4.1 and demonstrates the fact that it is not always possible to fit the scheduling within the planning period. The second example is more comprehensive and also considers non-zero initial WIP.

**Example A** Consider the manufacturing system introduced in Section 4.1, and its Petri net model shown in Figure 19. We will apply the scheduling approach to the second planning period, the duration of which is 40 time units. The number of times each transition has to fire during this period are given on Table 6 and are repeated below. For this first example we do not consider initial WIP. The initial marking is as shown in Figure 19.

Table 9: Transition firings for the second period

	Transition									
Period	$t_1$	$t_2$	$t_3$	$t_4$	$t_6$	$t_9$	$t_{10}$	$t_{12}$	$t_7$	$t_8$
$k = 2$	15	10	5	15	2	2	2	2	0	2

Step 1 of the algorithm produces the initial sequences presented below.

#### RU sequences

$$\sigma_{p_1} = < (t_2, 1), (t_3, 1), (t_3, 2), (t_3, 3), (t_3, 4),$$

$$(t_3, 5), (t_2, 2), (t_2, 3), (t_2, 4), (t_2, 5), (t_2, 6), (t_2, 7), (t_2, 8), (t_2, 9), (t_2, 10) >$$

$$\sigma_{p_4} = < (t_8, 1), (t_8, 2) >$$

## PS sequences

$$\sigma_{q_1} = < (t_2, 1), (t_9, 1), (t_9, 2), (t_2, 2), (t_2, 3), (t_2, 4),$$

$$(t_2, 5), (t_2, 6), (t_2, 7), (t_2, 8), (t_2, 9), (t_2, 10) >$$

$$\sigma_{q_2} = < (t_8, 1), (t_8, 2), (t_3, 1), (t_3, 2), (t_3, 3), (t_3, 4), (t_3, 5) >$$

$$\sigma_{q_3} = < (t_4, 1), (t_4, 2), (t_4, 3), (t_4, 4), (t_4, 5), (t_4, 6),$$

$$(t_4, 7), (t_4, 8), (t_4, 9), (t_4, 10), (t_4, 11), (t_4, 12), (t_4, 13), (t_4, 14), (t_4, 15) >$$

In Step 2 the critical path and the initial makespan are computed. The initial critical path comprises the following 20 transition firings:

$$CP_i = < (t_1, 1), (t_2, 1), (t_9, 1), (t_9, 2), (t_8, 2), (t_3, 1), (t_3, 2), (t_3, 3), (t_3, 4), (t_3, 5),$$

$$(t_2, 2), (t_2, 3), (t_2, 4), (t_2, 5), (t_2, 6), (t_2, 7), (t_2, 8), (t_2, 9), (t_2, 10), (t_4, 15) >$$

The makespan corresponding to this critical path is 56 time units. In steps 2, 3 and 4 a transition firing is selected from the makespan and is swapped with the previous transition firing in its corresponding sequence providing that the requirements of Step 3 are met. After the swap is performed the makespan and the critical path are recalculated. The procedure is continued until no further improvement is possible. The following list presents all swaps performed, along with the associated sequence and the resulting makespan.



$(t_9, 1) \Rightarrow (t_2, 1)$	$\sigma_{q_1}$	56
$(t_9, 2) \Rightarrow (t_2, 1)$	$\sigma_{q_1}$	56
$(t_3, 1) \Rightarrow (t_8, 2)$	$\sigma_{q_2}$	52
$(t_3, 1) \Rightarrow (t_2, 1)$	$\sigma_{p_1}$	52
$(t_3, 2) \Rightarrow (t_8, 2)$	$\sigma_{q_2}$	51
$(t_3, 2) \Rightarrow (t_2, 1)$	$\sigma_{p_1}$	51
$(t_3, 1) \Rightarrow (t_8, 1)$	$\sigma_{q_2}$	50
$(t_3, 3) \Rightarrow (t_8, 2)$	$\sigma_{q_2}$	50
$(t_3, 3) \Rightarrow (t_2, 1)$	$\sigma_{p_1}$	50

The final critical path comprises 14 transition firings:

$$CP_f = < (t_5, 1), (t_9, 1), (t_9, 2), (t_2, 1), (t_2, 2), (t_2, 3), (t_2, 4), (t_2, 5), \\ (t_2, 6), (t_2, 7), (t_2, 8), (t_2, 9), (t_2, 10), (t_4, 15) >$$

The corresponding makespan is 50 time units. Note, however, that the schedule does not fit within the planning period. Therefore production planning needs to be adjusted and the schedule should be recomputed for each planning period affected.

**Example B** In this example we will consider an FMS that comprises five machines  $M_1, \dots, M_5$  and manufactures 3 types of parts.

In Figure 22 the operating sequences for product types  $A, B$  and  $C$  are presented. Each block represents one operation and contains the alternative machines that can perform this operation. We assume that the modeling, design evaluation and process planning tasks have been performed as described in the previous sections. Figure 23 presents the Petri net model of the manufacturing system. Note that the initial marking of the model is not zero, modeling initial WIP. The correspondence between

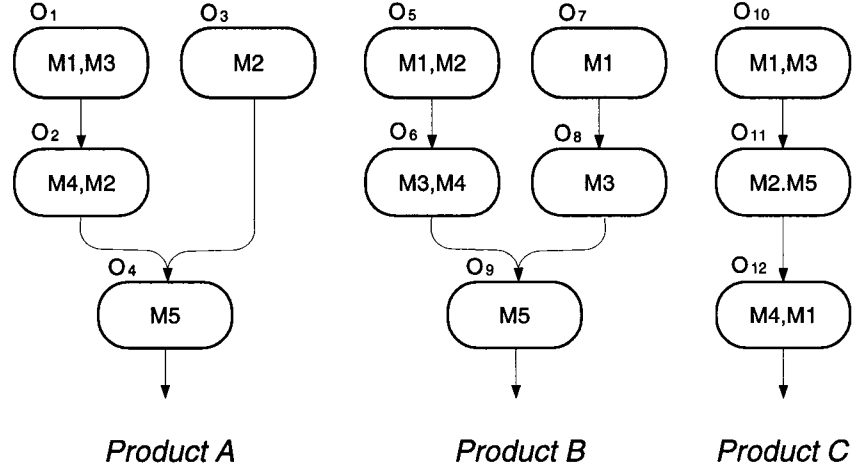


Figure 22: Manufacturing operations for the three part types of example B.

the machines of the system and transitions of the model is given by Table 10. For clarity, Figure 23 presents only place  $Q_1$ , which corresponds to the PS decisions related to machine  $M_1$ . Places  $Q_2, Q_3, Q_4$  and  $Q_5$  for machines  $M_1, \dots, M_5$  have been omitted. Also omitted are the places that prevent reentries on transitions  $t_1, t_6, t_9, t_{14}$  and  $t_{18}$ .

We will perform the the production scheduling for a plan period with duration 78 time units. Table 10 contains the firing times of the transitions and the number of times each transition fires during the time period under consideration.

By applying Step 1 of the scheduling algorithm, the following sequences are derived (for brevity only the transitions part of the firing is reported):

### RU sequences

$$\sigma_{p_1} = \langle t_3, t_3, t_2, t_2, t_2, t_3, t_3, t_3, t_3, t_3, t_3, t_3 \rangle$$

$$\sigma_{p_2} = \langle t_5, t_5, t_5, t_4, t_4, t_4, t_4, t_4, t_5, t_4, t_4, t_5 \rangle$$

$$\sigma_{p_{12}} = \langle t_{20}, t_{19}, t_{19}, t_{20}, t_{19}, t_{20}, t_{20}, t_{19}, t_{19}, t_{20}, t_{19}, t_{19}, t_{19}, t_{19}, t_{19}, t_{19}, t_{19}, t_{19} \rangle$$

$$\sigma_{p_{13}} = \langle t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21} \rangle$$

$$\sigma_{p_{14}} = \langle t_{23}, t_{23}, t_{24}, t_{23}, t_{23}, t_{24}, t_{23}, t_{23}, t_{24}, t_{23}, t_{23}, t_{23}, t_{23}, t_{23}, t_{23}, t_{23}, t_{23}, t_{23}, t_{23} \rangle$$

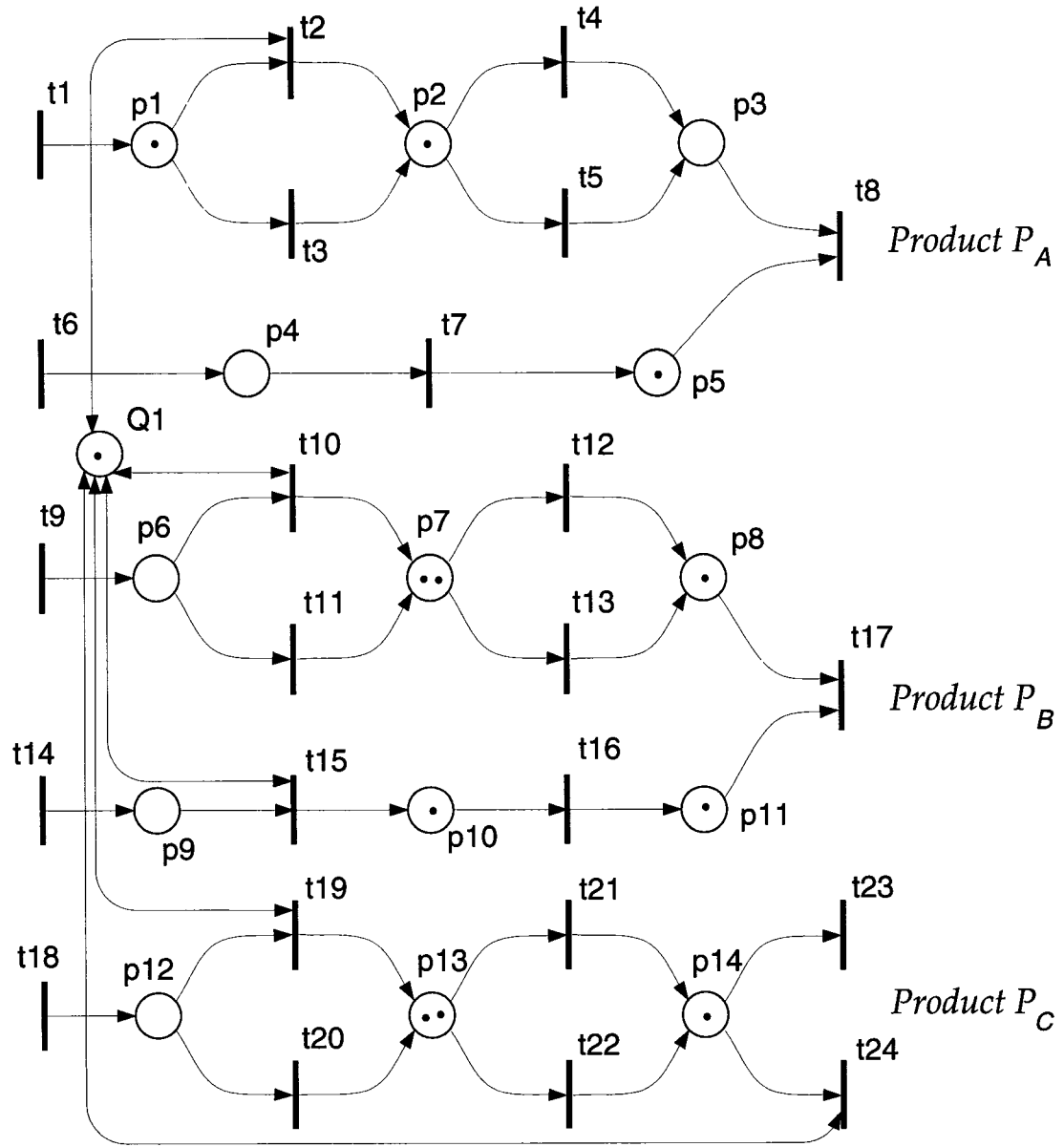


Figure 23: Manufacturing process model.

Table 10 Transition information for example B

Transition	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_{10}$	$t_{11}$	$t_{12}$
Firing time	0	1	3	2	4	0	2	4	0	3	1	8
$n_t$	12	3	9	7	5	12	12	12	11	0	11	0
Machine	—	$M_1$	$M_3$	$M_2$	$M_4$	—	$M_2$	$M_5$	—	$M_1$	$M_2$	$M_3$
Operation	—	$O_1$	$O_1$	$O_2$	$O_2$	—	$O_3$	$O_4$	—	$O_5$	$O_5$	$O_6$

Transition	$t_{13}$	$t_{14}$	$t_{15}$	$t_{16}$	$t_{17}$	$t_{18}$	$t_{19}$	$t_{20}$	$t_{21}$	$t_{22}$	$t_{23}$	$t_{24}$
Firing time	1	0	2	3	2	0	2	2	1	1	2	1
$n_t$	11	11	11	11	11	20	15	5	20	0	17	3
Machine	$M_4$	—	$M_1$	$M_3$	$M_5$	—	$M_1$	$M_3$	$M_2$	$M_5$	$M_4$	$M_1$
Operation	$O_6$	—	$O_7$	$O_8$	$O_9$	—	$O_{10}$	$O_{10}$	$O_{11}$	$O_{11}$	$O_{12}$	$O_{12}$

### PS sequences

$$\sigma_{q_1} = < t_{19}, t_{15}, t_2, t_{19}, t_{15}, t_{15}, t_{24}, t_2, t_{19}, t_{15}, t_{15}, t_2, t_{15}, t_{15}, t_{15}, t_{24}, t_{15}, t_{19},$$

$$t_{15}, t_{15}, t_{24}, t_{19}, t_{19}, t_{19}, t_{19}, t_{19}, t_{19}, t_{19}, t_{19}, t_{19}, t_{19}, t_{19} >$$

$$\sigma_{q_2} = < t_3, t_{20}, t_3, t_{16}, t_{16}, t_{20}, t_{16}, t_{16}, t_{16}, t_{16}, t_{16}, t_{16}, t_{20}, t_3, t_3, t_{16}, t_{20}, t_3, t_3, t_{16},$$

$$t_{20}, t_{16}, t_3, t_3, t_3 >$$

$$\sigma_{q_3} = < t_{11}, t_7, t_{11}, t_7, t_{21}, t_{11}, t_4, t_{21}, t_4, t_7, t_4, t_7, t_7, t_{11}, t_{11}, t_4, t_{11}, t_{21}, t_{21}, t_{21},$$

$$t_4, t_{21}, t_{11}, t_{11}, t_{21}, t_7, t_{11}, t_{11}, t_4, t_{21}, t_{11}, t_{21}, t_{21}, t_{21}, t_7, t_7, t_7, t_7, t_7,$$

$$t_7, t_4, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21} >$$

$$\sigma_{q_4} = < t_{23}, t_5, t_5, t_{13}, t_{13}, t_{13}, t_{23}, t_{13}, t_5, t_{13}, t_{13}, t_{13}, t_{13}, t_{13}, t_{13}, t_{23},$$

$$t_{23}, t_{23}, t_{13}, t_5, t_{23}, t_{23}, t_{23}, t_{23}, t_{23}, t_{23}, t_{23}, t_5, t_{23}, t_{23}, t_{23}, t_{23}, t_{23} >$$

$$\sigma_{q_6} = < t_{17}, t_8, t_8, t_{17}, t_{17}, t_8, t_8, t_8, t_8, t_{17}, t_{17}, t_{17}, t_{17}, t_{17}, t_{17}, t_{17},$$

$$t_8, t_{17}, t_8, t_8, t_8, t_8, t_8 >$$

In Step 2 the initial makespan and the critical path are calculated:

$$\begin{aligned}
 CP_i = & \langle (t_{23}, 1), (t_5, 1), (t_5, 2), (t_{13}, 1), (t_{13}, 2), (t_{13}, 3), (t_{23}, 2), (t_{13}, 4), (t_5, 3), (t_4, 1), (t_{21}, 2), \\
 & (t_{24}, 1), (t_2, 2), (t_{19}, 3), (t_{15}, 4), (t_{16}, 5), (t_{16}, 6), (t_{16}, 7), (t_{16}, 8), (t_{20}, 3), (t_3, 3), (t_4, 4), \\
 & (t_{11}, 6), (t_{21}, 3), (t_{21}, 4), (t_{21}, 5), (t_4, 5), (t_{21}, 6), (t_{11}, 7), (t_{13}, 9), (t_{13}, 10), (t_{23}, 3), (t_{23}, 4), \\
 & (t_{23}, 5), (t_{13}, 11), (t_5, 4), (t_{23}, 6), (t_{24}, 3), (t_{19}, 5), (t_{21}, 11), (t_7, 7), (t_7, 8), (t_7, 9), (t_7, 10), \\
 & (t_7, 11), (t_7, 12), (t_4, 7), (t_{21}, 12), (t_{23}, 10), (t_{23}, 11), (t_{23}, 12), (t_5, 5), (t_{23}, 13), (t_{23}, 14), \\
 & (t_{23}, 15), (t_{23}, 16), (t_{23}, 17) \rangle
 \end{aligned}$$

The critical path has 57 transition firings and the makespan is 104 time units.

Subsequently steps 2, 3 and 4 are performed, until there are no more pairs of the transition firings to be swapped in step 4, and the algorithm stops. Due to space constraints only some of the initial and final swaps are presented below. The list presents the associated sequence, the firing of the critical path that was improved and the resulting makespan.

1	$\sigma_{q_4}$	$(t_5, 1)$	104
2	$\sigma_{q_4}$	$(t_5, 2)$	104
3	$\sigma_{q_4}$	$(t_{13}, 1)$	104
4	$\sigma_{q_4}$	$(t_{13}, 1)$	104
5	$\sigma_{q_4}$	$(t_{13}, 1)$	104
$\vdots$	$\vdots$	$\vdots$	$\vdots$
249	$\sigma_{q_2}$	$(t_{20}, 4)$	80
250	$\sigma_{q_4}$	$(t_{23}, 7)$	79
251	$\sigma_{q_3}$	$(t_{21}, 8)$	79
252	$\sigma_{q_3}$	$(t_{21}, 17)$	79
253	$\sigma_{q_3}$	$(t_{11}, 11)$	78

After 253 steps the makespan has been reduced adequately and fits with the planning period. The algorithm is interrupted. The final *RU* and *PS* place sequences and the

critical path are given below.

### RU sequences

$$\sigma_{p_1} = \langle t_3, t_3, t_2, t_2, t_2, t_3, t_3, t_3, t_3, t_3, t_3 \rangle$$

$$\sigma_{p_2} = \langle t_5, t_5, t_4, t_5, t_4, t_4, t_4, t_4, t_5, t_4, t_5, t_4 \rangle$$

$$\sigma_{p_{12}} = \langle t_{19}, t_{19}, t_{20}, t_{20}, t_{19}, t_{20}, t_{20}, t_{19}, t_{19}, t_{19}, t_{19}, t_{20}, t_{19}, t_{19}, \\ t_{19}, t_{19}, t_{19}, t_{19}, t_{19}, t_{19} \rangle$$

$$\sigma_{p_{13}} = \langle t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, t_{21}, \\ t_{21}, t_{21}, t_{21}, t_{21}, t_{21} \rangle$$

$$\sigma_{p_{14}} = \langle t_{24}, t_{23}, t_{23}, t_{23}, t_{24}, t_{23}, t_{23}, t_{24}, t_{23}, t_{23}, t_{23}, t_{23}, t_{23}, t_{23}, \\ t_{23}, t_{23}, t_{23}, t_{23}, t_{23} \rangle$$

### PS sequences

$$\sigma_{q_1} = \langle t_{19}, t_{15}, t_2, t_{19}, t_{15}, t_{15}, t_{24}, t_2, t_{19}, t_{15}, t_{15}, t_2, t_{15}, t_{15}, t_{15}, t_{24}, \\ t_{15}, t_{19}, t_{15}, t_{15}, t_{24}, t_{19}, t_{19}, t_{19}, t_{19}, t_{19}, t_{19}, t_{19}, t_{19}, t_{19} \rangle$$

$$\sigma_{q_2} = \langle t_3, t_3, t_{20}, t_{20}, t_{16}, t_{16}, t_{20}, t_{20}, t_{16}, t_{16}, t_{16}, t_{16}, t_{16}, t_{16}, t_3, t_3, \\ t_{16}, t_3, t_3, t_{16}, t_{20}, t_{16}, t_3, t_3, t_3 \rangle$$

$$\begin{aligned}
\sigma_{q_3} = & < t_{11}, t_7, t_{11}, t_7, t_{21}, t_{11}, t_4, t_{21}, t_4, t_7, t_4, t_7, t_7, t_{11}, t_{11}, t_{11}, t_{21}, \\
& t_{21}, t_{21}, t_{21}, t_{11}, t_{11}, t_{21}, t_7, t_{11}, t_{11}, t_{21}, t_4, t_{11}, t_4, t_{21}, t_{21}, t_{21}, t_{21}, \\
& t_7, t_4, t_7, t_{21}, t_{21}, t_{21}, t_{21}, t_7, t_7, t_7, t_{21}, t_7, t_4, t_{21}, t_{21}, t_{21} > \\
\sigma_{q_4} = & < t_5, t_{13}, t_{13}, t_{13}, t_{13}, t_5, t_{23}, t_{23}, t_5, t_{13}, t_{13}, t_{13}, t_{13}, t_{13}, t_{13}, t_{23}, \\
& t_{23}, t_{23}, t_{13}, t_{23}, t_{23}, t_5, t_{23}, t_{23}, t_{23}, t_{23}, t_{23}, t_5, t_{23}, t_{23}, t_{23}, t_{23}, t_{23} > \\
\sigma_{q_6} = & < t_{17}, t_8, t_8, t_{17}, t_{17}, t_8, t_8, t_8, t_8, t_{17}, t_{17}, t_{17}, t_{17}, t_{17}, t_{17}, t_{17}, \\
& t_8, t_8, t_{17}, t_8, t_8, t_8, t_8 >
\end{aligned}$$

The final critical path has 30 transition firings. The makespan duration is 78 time units.

$$\begin{aligned}
CP_f = & < (t_3, 1), (t_3, 2), (t_{20}, 1), (t_{20}, 2), (t_{16}, 1), (t_{16}, 2), (t_{20}, 3), (t_{20}, 4), (t_{16}, 3), (t_{16}, 4), \\
& (t_{16}, 5), (t_{16}, 6), (t_{16}, 7), (t_{16}, 8), (t_3, 3), (t_3, 4), (t_{16}, 9), (t_3, 5), (t_5, 4), (t_{23}, 8), (t_{23}, 9), \\
& (t_{23}, 10), (t_{23}, 11), (t_{23}, 12), (t_5, 5), (t_{23}, 13), (t_{23}, 14), (t_{23}, 15), (t_{23}, 16), (t_{23}, 17) >
\end{aligned}$$

Figure 24 presents the Ghant chart of the firings of each transition. Transitions with zero firing time, and transitions that do not fire have been omitted.

## 4.5 Software Implementation

All procedures described in this chapter have been implemented in a software system which i) analyzes a CFIO net in order to determine whether the net can be kept bounded, and whether it is consistent, ii) performs production planning given the demand and associated costs; iii) performs the production scheduling given the planning results.

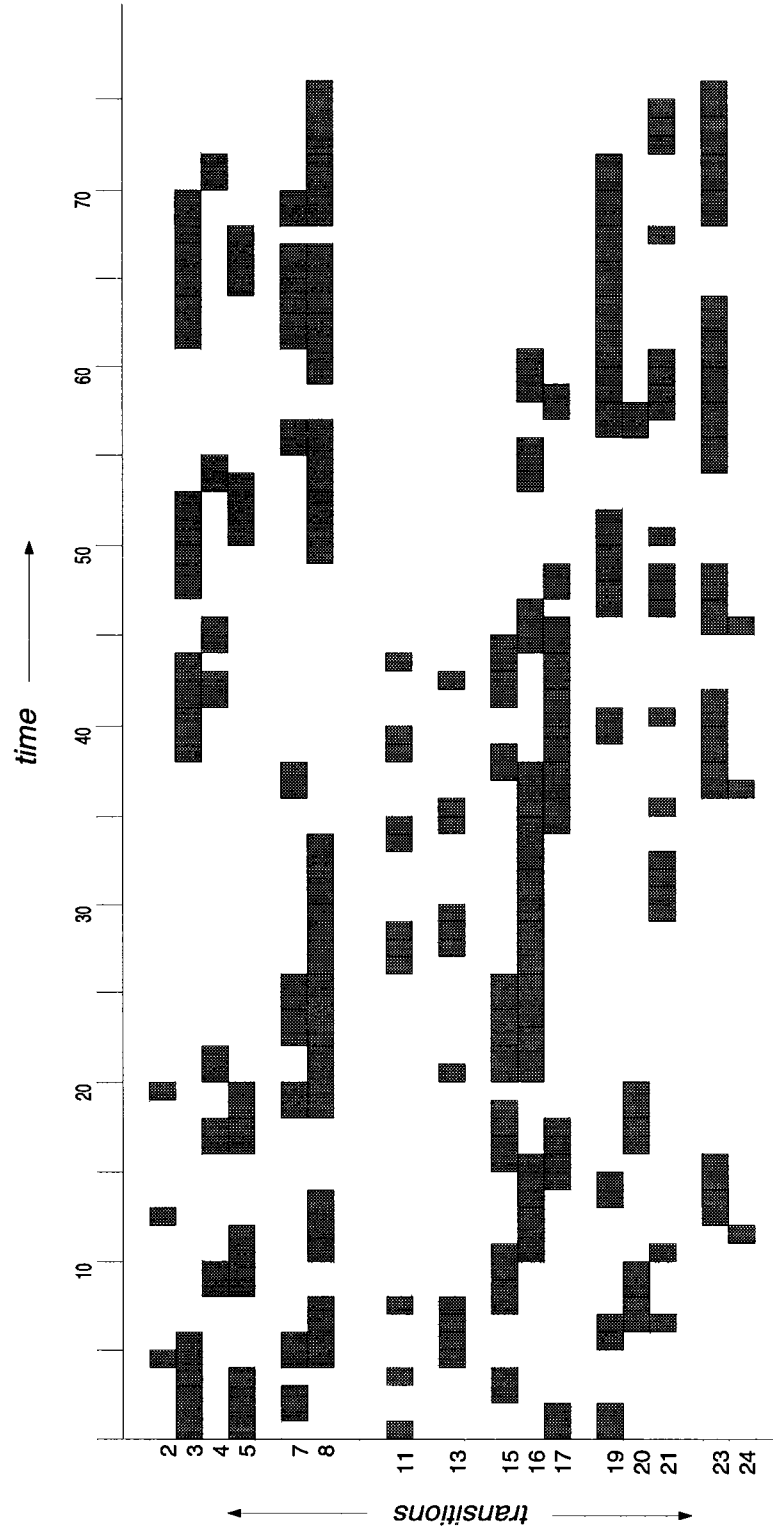


Figure 24: Scheduling results



The software accepts Petri nets in the form of an incidence matrix and performs the following steps:

1. Design Analysis:

- a. Circuit reduction: The input is the incidence matrix of a single CFIO net. If no inconsistent or self-sustained circuits exist, all circuits are reduced to single transitions. Otherwise, an appropriate error message is generated. The output of the circuit reduction is the incidence matrix of the reduced CFIO net.
- b. Path reduction: This procedure accepts as input the output incidence matrix provided in the output of the previous step, and reduces it according to the path reduction method. The output of this step is the incidence matrix of the reduced net.
- c. Consistency detection: The input of this step is the output incidence matrix of the path reduction step. The consistency of the net is examined by calculating the extremal solutions of  $C^1 x^1 = 0$ , where  $C^1$  is the incidence matrix of the reduced net.

2. Production Planning: The linear program as described in Section 4.3.1 is formulated and solved.

3. Scheduling: It accepts as input the incidence matrix of the Petri net model. It is assumed that the model is decomposable to manageable CFIO nets. Given an initial marking, the software calculates via simulation the initial schedule and the starting times of all transition firings. Secondly, the critical path and float times are calculated. Subsequently transition firings are swapped if such swapping does not

increase the makespan and the schedule is recalculated. Finally the final makespan, the critical path and the transition firings times are reported.

The software has been implemented in the C programming language.

## **4.6 Chapter Summary**

In this chapter we discussed the modeling of an FMS using Petri nets that are decomposable to CFIO nets, and we applied the methods introduced in Chapter 3 to evaluate the design of the system with respect to manageability. Subsequently, we introduced a production planning approach that uses a CFIO net decomposition of the system model to formulate the corresponding linear program. The latter when solved provides, for each planning time period, the number of times each operation should be performed and by which machine. If the Petri net model is decomposed to minimal CFIO nets, the flexibility of control is maximal but the computational burden increases. Finally, we developed a critical path-based scheduling algorithm that supports two types of decisions: resource usage and product sequencing decisions. The algorithm provides, for each planning period, the time each operation should be performed, so that the makespan fits within the planning period (if possible).

## 5 Conclusions

In this work we developed a Petri net-based model, the Conflict Free nets with Input and Output transitions (CFIO) nets, to support the modeling and verification of FMS designs, as well as production planning and scheduling during FMS operation.

CFIO nets are used to detect important qualitative properties of the system such as i) liveness, which guarantees that the FMS is deadlock free, ii) boundedness, which guarantees that the work-in-process of the FMS can be held below a given level, and iii) consistency, which guarantees that any valid inventory level, as well as any machine availability state, can be reached. CFIO nets are used to formulate the production planning problem. They facilitate problem abstraction in a convenient manner so that to strike a desirable balance between solution quality and computational burden. Having the production plan, production scheduling is performed using the Petri net model, to obtain the time each operation needs to be performed so that the makespan of the schedule fits within each planning period.

The contributions of this work can be summarized as follows:

- (i) Introduced a special class of conflict-free Petri nets (CFIO nets) which facilitates modeling and analysis of flexible manufacturing systems.
- (ii) Established the qualitative properties of CFIO nets.
- (iii) Developed efficient procedures to assess these qualitative properties of a given CFIO net.
- (iv) Employed the Petri net model of an FMS to formulate the production planning problem.

- (v) Introduced a critical path based-scheduling procedure that supports the two types of decisions to be made while scheduling the FMS.

It is noted that the production planning and scheduling methods presented in this work do not represent the only possible approaches, but rather serve as a model for the type of analysis that the CFIO net formulation can support. Further study is needed to examine the implementation of other optimization criteria in the planning method (e.g. machine utilization) and other approaches for the scheduling problem (e.g. the shifting-bottleneck procedure [8] and global scheduling with genetic algorithms [14], etc.). Further research could also improve our current scheduling heuristic, by allowing rescheduling of operations that would result in temporarily increasing the makespan, so that local minima could be avoided in the search for a global optimum.

## 6 Bibliography

- [1] B.H. Krogh, and C.L. Beck, "Synthesis of Place/Transition nets for Simulation and Control of Manufacturing Systems," *Proceedings of IFIP Symp. Large Scale System*, Zurich, Switzerland, August 1986.
- [2] Laftit, S., Proth, J.-M. and Xie, X. L., "Optimization of Invariant Criteria for Event Graphs," *IEEE Transaction on Automatic Control*, vol. 37, no. 5, pp. 547–555, May 1992.
- [3] Proth, J.-M., and Vernadat, F., "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.
- [4] Lautenbach, K., "Linear Algebraic Techniques for PT Nets," *Lecture Notes in Computer Science — Advanced in Petri Nets 1986*, vol. 1, pp. 142, 1986.
- [5] Hu, T.C., *Integer Programming and Networks Flow*. Mass.: Addison Wesley Reading, 1969.
- [6] Memmi, G., "Linear Algebra in Net Theory," *Lecture Notes in Computer Science — Net Theory and Applications*, pp. 213–223, 1979.
- [7] Sifakis, J., "Structural Properties of Petri Nets," *Lecture Notes in Computer Science — Mathematical Foundations of Computer Science*, pp. 474–483, 1978.
- [8] Adams, J., Balas, E., and Zawack, D., "The shifting bottleneck procedure for job shop scheduling," *Management Science*, Vol. 34, pp. 391–401, 1988

- [9] Baker, K.R., *Elements of Sequencing and Scheduling*, Amos Tuck School of Business Administration, Dartmouth College, Hanover, New Hampshire, 1992.
- [10] Barker, J. R., and McMahon, G. B., "Scheduling the general job shop," *Management Science*, Vol. 31, pp. 594-598, 1985.
- [11] Carlier, J., and Pinson, E., "An algorithm for solving the job shop problem," *Management Science*, Vol. 35, pp. 164-176, 1989.
- [12] McCarthy, B. L., Jiyan Liu, "Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling," *International Journal of Production Research*, Vol. 31, No. 1, pp. 59-79, 1993
- [13] Greenberg, H. H., "A branch and bound solution to the general scheduling problem," *Operations Research*, Vol. 16, pp. 353-361, 1968.
- [14] Herrmann, J.W., Lee, C.-Y. and Snowdon, J.L., "A classification of static scheduling problems," *Complexity in Numerical Optimization*, P.M.Pardalos, ed., pp.203-253, World Scientific, River Edge, New Jersey, 1993.
- [15] Hoitomt D.J., Luh P.B. *et al.*, "Schedule Generation and Reconfiguration for Parallel Machines," *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 6, pp.687-696, 1990.
- [16] Hoitomt D.J., Perkins J.B. and Luh P.B., "Distributed Scheduling of Job Shops," *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, CA, pp. 1067-1072, April 1991.
- [17] Lageweg, B. J., Lenstra, J. K., and Rennooy Kan, A. H. G., "Job shop scheduling by implicit enumeration," *Management Science*, Vol. 24, pp. 441- 450, 1977.

- [18] Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G., Shmoys D.B., "Sequencing and Scheduling Algorithms and Complexity," *Handbooks in Operations and Management Science*, Vol. 4: *Logistics of Production and Inventory*, North-Holland Pub. 1985.
- [19] Miltenburg, J., and Sinnamon, G., "Scheduling Mixed-Model Multi-Level Just-In-Time Production Systems," *International Journal of Production Research*, Vol. 27, No. 9, pp. 1487-1509, 1989.
- [20] Miltenburg, J., and Sinnamon, G., "Algorithms for Scheduling Multi-Level Just-In-Time Production Systems," *IIE Transactions*, Vol. 24, No. 2, pp. 121-130, 1992
- [21] Owens T. and Luh P.B., "A Job Completion Time Estimation Method for Work Center Scheduling," *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, CA, pp.110–115, April 1991.
- [22] Panwalkar, S. S., and Iskander, W., "A Survey of Scheduling Rules," *Operations Research*, Vol. 25, No. 1, pp. 45-61, 1977
- [23] Rodammer, Frederick A., K. Preston White, Jr., "A Recent Survey of Production Scheduling," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 18, No. 6, pp. 841-851, 1988
- [24] Zahorik, A., Thomas L. J., and Trigeiro W., "Network Programming Models for Production Scheduling in Multi-Stage," *Multi-Item Capacitated Systems, Management Science*, Vol. 30, No. 3, pp. 308-325, 1984.
- [25] Al-Jaar R.Y., Desrochers A., "A survey of Petri nets in automated manufacturing systems," *IMACS World Congress*, Vol 2, pp. 503-510, Paris, June 1988.

- [26] Silva M., “Sur le concept de macroplace et son utilisation pour l’analyse des réseaux de Petri.” *RAIRO-Systems Analysis and Control*, 15 (4), pp. 57-67, 1981.
- [27] Silva M., Colom J.M., “On the computation of structural synchronic invariants in P/T nets,” *Advances in Petri Nets* 1988 (G. Rozenberg, ed.). LNCS 340, pp. 386-417, Springer-Verlag, Berlin, 1988.
- [28] Berthelot G., “Checking properties of nets using transformations” *LNCS, Advances in Petri Nets*, Vol. 222 pp.19–40, NY, Springer Verlag, 1985
- [29] Brauer W., Reisig W., Rozenberg G., “Petri nets: Central models and their properties,” *Lecture Notes in Computer Science* 254, Springer-Verlag, Berlin, 1987.
- [30] Colom J.M., Silva M., “Convex geometry and semiflows in P/T nets,” *Advances in Petri Nets* 1990 (G. Rozenberg, ed.). LNCS 483, Springer-Verlag, Berlin, pp. 79-112, 1991.
- [31] Rozenberg G. (eds.), *Advances in Petri nets. Lecture Notes in Computer Science*: 188 (APN 84), 222 (APN 85), 266 (APN 87), 340 (APN 88), 424 (APN 89) and 483 (APN 90). Springer-Verlag, Berlin, various years.
- [32] Carlier J., Chretienne P., “Timed Petri Net Schedules,” *LNCS 340, Advances in Petri Nets*, 1988.
- [33] DiCesare F., Harhalakis G., Proth J.-M., Silva M., Vernadat F., *Practice of Petri Nets in Manufacturing* (Andover, England: Chapman & Hall Ltd, 1993).
- [34] Garey, M.R., and Johnson D.S., *Computers and intractability: a Guide to the Theory of NP-Completeness*, W.H. Freeman, San Francisco, 1979.



- [35] Harhalakis G., Leventopoulos M., Lin C.-P., Nagi R. Proth J.-M., "A Class of Conflict-Free Petri Nets used for controlling Manufacturing Systems", *INRIA-Lorraine, Rapport de Recherche* No. 1908, May 1993.
- [36] Knuth E.D., *The Art of Computer Programming*, Vol 1: *Fundamental Algorithms* (Reading, MA: Addison-Wesley, 1973 [2nd ed.]) §2.2.3 .
- [37] Lee D.Y. and DiCesare F., "FMS Scheduling Using Petri Nets and Heuristic Search," *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Nice, France, May 1992.
- [38] Di Mascolo M., Frein Y., Dallery Y., David R., "A unified modeling of Kanban systems using Petri nets," *Technical Report No. 89-06*, LAG. Grenoble, France, September 1989.
- [39] Hillion H.P., Proth J.M., "Performance evaluation of job-shop systems using timed event-graphs," *IEEE Transactions on Automatic Control*, Vol. 34, No. 1, pp. 3-9, 1989.
- [40] Murata T., "Petri nets: properties, analysis and applications," *Proceedings of the IEEE*, Vol. 77, No. 4, April 1989.
- [41] Ramamoorthy C.V., Ho G.S., "Performance evaluation of asynchronous concurrent systems using Petri nets," *IEEE Transactions, Software Engineering.*, Vol. SE-6, No 5, pp. 440-449, 1980.
- [42] Ramchandani C., "Analysis of asynchronous concurrent systems by timed Petri nets," *Lab. Comput. Sci., Mass. Inst. Tech.* Cambridge, MA, Technical Report 120, 1974.
- [43] Shih H.M., Sekiguchi T., "A Timed Petri Net and Beam Search Based On-Line

- FMS Scheduling System with Routing Flexibility”, *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, CA, April 1991.
- [44] Silva M., Valette R. “Petri nets and Flexible Manufacturing”. *Advances in Petri Nets* 1989 (G. Rozenberg, ed.). LNCS 424, pp. 375-417. Springer-Verlag, Berlin, 1990.
- [45] Stoner, R.H., Wu, S.Y.D., Vaccari, R., “New search spaces for sequencing problems with application to job shop scheduling,” *Management Science*, Vol.38, No.10, pp. 1495–1505, 1992.
- [46] Sakurai M., “The Influence of Factory Automation on Management Accounting Practices,” *Measures for Manufacturing Excellence*, (Harvard Business School Press, Boston, MA, 1990)
- [47] *The Flexible Manufacturing Systems Handbook*, The Charles Stark Draper Laboratory, Cambridge, Massachusetts, 1982
- [48] Kusiak a., “Flexible Manufacturing Systems: A Structural Approach,” *Int.J. Prod. Res.*, vol. 23, no. 6, pp. 1057–1073, 1985
- [49] Suri R., Whitney C., “Decision Support Requirements in Flexible Manufacturing,” *J. of Manuf. Syst.*, vol. 3, no. 1, pp 27–38, 1984.
- [50] Blazewicz J., Finke G., Haupt R., Schmit G., “New Trends in Machine Scheduling,” *European J.Oper.Res.* vol 37, pp 303–317, 1983