

Continuous Approach of Scheduling Problems Based on Petri Nets

Jean-Marie Proth, Nathalie Sauer

► **To cite this version:**

Jean-Marie Proth, Nathalie Sauer. Continuous Approach of Scheduling Problems Based on Petri Nets. [Research Report] RR-2822, INRIA. 1996, pp.16. inria-00073870

HAL Id: inria-00073870

<https://hal.inria.fr/inria-00073870>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

*Continuous Approach of Scheduling
problems Based on Petri Nets*

Jean-Marie Proth - Nathalie Sauer

N° 2822

Mars 1996

THÈME 4

R
apport
de recherche

Les rapports de recherche de l'INRIA
sont disponibles en format postscript sous
ftp.inria.fr (192.93.2.54)

si vous n'avez pas d'accès ftp
la forme papier peut être commandée par mail :
e-mail : dif.gesdif@inria.fr
(n'oubliez pas de mentionner votre adresse postale).

par courrier :
Centre de Diffusion
INRIA
BP 105 - 78153 Le Chesnay Cedex (FRANCE)

INRIA research reports
are available in postscript format
ftp.inria.fr (192.93.2.54)

if you haven't access by ftp
we recommend ordering them by e-mail :
e-mail : dif.gesdif@inria.fr
(don't forget to mention your postal address).

by mail :
Centre de Diffusion
INRIA
BP 105 - 78153 Le Chesnay Cedex (FRANCE)

Continuous approach of scheduling problems based on Petri nets

Jean-Marie PROTH ^{a,b}, Nathalie SAUER ^a

Abstract:

Scheduling a manufacturing system is usually an NP-hard problem. This means that only heuristic algorithms can be used to provide near-optimal schedules. In this paper, we show that a manufacturing system can be modelled using a particular type of Petri nets, called Controllable-Output nets, or CO nets for short. These Petri net models are then used to introduce a two-stage scheduling algorithm for large horizon problems. The first stage consists of distributing the workload among the resources. The second stage derives a schedule from the resource workload. The deterministic case is considered. Numerical results are proposed.

Keywords: Flexible Manufacturing Systems, Scheduling Theory, Petri nets, Planning.

Une approche continue des problèmes d'ordonnancement basée sur les réseaux de Petri

Jean-Marie PROTH ^{a,b}, Nathalie SAUER ^a

Résumé :

L'ordonnancement des systèmes de production est un problème NP difficile. Par conséquent, il est nécessaire d'utiliser des heuristiques pour obtenir des solutions proches de la solution optimale. Dans ce papier, nous montrons que les systèmes de production peuvent être modélisés à l'aide d'une classe particulière de réseaux de Petri, appelés réseaux à sorties contrôlables ou CO nets (CO signifie "Controllable-Output"). Cette modélisation permet d'introduire un algorithme d'ordonnancement en deux étapes pour les problèmes sur un horizon important. La première étape consiste à déterminer la charge des différentes machines. La seconde étape permet d'ordonner les ressources en utilisant les résultats de la première étape. Le cas déterministe est considéré dans ce papier. Des résultats numériques sont proposés pour valider les résultats obtenus.

Mots clés : Systèmes de production, Ordonnancement, Planification, Réseaux de Petri.

^a INRIA-Lorraine, Technopôle Metz 2000, 4 rue Marconi, 57070 Metz, FRANCE

^b Institute for Systems Research, University of Maryland, College Park, MD 20742, USA

1. Introduction

Petri nets have been recognized as a powerful tool for the specification, the modeling and the evaluation of discrete event systems, especially manufacturing systems. An excellent survey on the subject can be found in Murata [6]. Silva and Valette [12] as well as Di Cesare et al. [2] present a thorough analysis on the subject.

As far as modeling and evaluation are concerned, researchers usually classify the manufacturing systems in two types, namely the cyclic manufacturing systems, in which production rates are constant, and non cyclic manufacturing systems, in which decisions are made to optimally meet customers' requirements.

Cyclic manufacturing systems have been, widely studied in Hillion and al. [3], Laftit and al. [4] and Proth and al. [9] from an analytical point of view. These authors showed that a cyclic manufacturing system can be modeled as an event graph. They also proposed several heuristic algorithms, as well as a branch and bound approach, to maximize the throughput while minimizing a linear combination of the markings. These approaches have been developed in the deterministic case, i.e. in the case when the firing times of the transitions are deterministic. Further researches have been conducted in the case when the firing times are stochastic: for more information, see Proth and al. [7-8]. Note that the basic results used to develop the research on the deterministic case are due to Commoner and al. [1] and Ramamoorthy and al. [11], to quote only the most important ones. Note also that perturbation analysis is widely used in the stochastic case. For more information on perturbation analysis, see for instance Suri [13] and the related references.

The goal when analysing non cyclic manufacturing systems is to meet given customers' requirements while minimizing Work-In-Process (WIP). Since the size of non cyclic manufacturing systems is usually very large, as compared to cyclic manufacturing systems, a hierarchical production management approach is commonly used to solve this kind of problems. The literature on the subject is huge. In the hierarchical framework, the global decision marking process is decomposed into a series of subproblems that correspond to different levels of the hierarchy. The solution of a subproblem imposes constraints on the solution of the subsequent lower level problem. The number of hierarchical levels depends on the complexity of the global problem. The lowest level of the hierarchy is the scheduling level. The goal at this level is to make the decisions on how to perform the tasks which are the solution of the previous level, which is the short term planning level. These tasks should be performed by a given instant called the horizon of the scheduling problem.

The short term planning and the scheduling problems can be set using a special type of Petri nets called CO nets, where CO stands for controllable output nets (see Proth and al. [10]). A powerful heuristic, based on the properties of the CO nets, has already been developed by Leventopoulos [5].

In this paper, we assume that:

- (i) customers' requirements are huge and given by ratios which may change from one day to the next one,
- (ii) the types of requirements are always the same,
- (iii) the goal is to maximize the throughput.

The changes in the ratios are made in order to adjust the production to the flow of demands. This situation arises, in particular, in food industry where the goal is to maximize the production while adjusting the ratios among products of the same family, according to selling objectives which are defined every day.

Section 2 introduces the CO nets and shows how to use these nets to model non cyclic manufacturing systems. The problem is formulated in section 3. In section 4, we set the short planning problem and present the related continuous formulation. In section 5, we show that the optimal throughput is reachable using a constrained operating mode (COM), and we propose a scheduling algorithm which leads to the optimal throughput. Numerical examples are given in section 6. Section 7 is the conclusion.

2. Modeling a manufacturing system

The job-shop model presented in this section is based on a new type of Petri nets called Controllable-Output nets, or CO nets for short. CO nets have been first introduced by Proth and al. [10] We give the definition of the CO nets in the next sub-section, and then show how to use these CO nets to model manufacturing systems.

2.1. Controllable nets

A controllable net is a net having two types of places: the process places and the resource places. The process places contain tokens which represent part while the resource places contain tokens which represent resources. In this paper, the only resources used are the machines.

Formally, a CO net is denoted by $G = (P \cup R, T, F, M_0)$, where P is the set of process places, R is the set of resource places, T is the set of transitions, F is the set of arcs, and M_0 is the initial marking.

G is a CO net if the following conditions hold:

- (K1) $\forall t \in T, \exists r \in R$ s.t. $(t,r) \in F$ and $(r,t) \in F$. Thus, for each transition, there exists a resource place which is an input place and an output place of the transition.
- (K2) $M_0(r) \geq 1, \forall r \in R$. Each resource place contains initially at least one token. In this paper, we assume that each resource place contains exactly one token.
- (K3) The subnet $G' = (P, T, F', M'_0)$, where F' is the restriction of F to $(P \times T) \cup (T \times P)$ and M'_0 is the restriction of M_0 to the process places, is an acyclic graph. We will see that this acyclic graph is not a connected graph in our application.
- (K4) There exists $T_I \subset T$ s.t. $t^0 = \emptyset, \forall t \in T_I$, and $T_O \subset T$ s.t. $t^0 = \emptyset, \forall t \in T_O$.

Furthermore:

- if $o_t = \emptyset$, $t^o \neq \emptyset$,
- if $t^o = \emptyset$, $o_t \neq \emptyset$,
- $o_p \neq \emptyset$ and $p^o \neq \emptyset$, $\forall p \in P \cup R$.

We denote by o_t (resp. o_p) the set of input places (resp. transitions) of t (resp. p). We also denote by t^o (resp. p^o) the set of output places (resp. transitions) of t (resp. p).

T_I is the set of input transitions and T_O the set of output transitions of G .

(K5) There exists a set W of t -invariants of G' such that:

- (i) $\forall t \in T_O, \exists \tau \in W$ s.t. $t \in \tau$ and $\sim t' \neq t, t' \in \tau$ s.t. $t' \in T_O$. In other words, it is possible to associate a t -invariant to each output transition, and this t -invariant contains only one output transition.
- (ii) $\{t / t \in W\} = T$ and $\{p / p \in W\} = P$: the set of t -invariants belonging to W covers G' .

It is easy to prove that a CO net is consistent, live, reversible, and can be kept bounded (see Proth et al. [10]).

2.2. CO nets for modeling manufacturing systems

A manufacturing process can be defined as follows:

- a component is obtained by performing a sequence of operations on raw material or on a component obtained by assembling other components,
- an assembly operation derive a component from several other components,
- a finished product is a component which is not further assembled or transformed.

Figure 1 represents a manufacturing process. In this process, raw material H_1 is transformed into component C_1 by performing successively operations O_1 and O_2 , and raw material H_2 is transformed into component C_2 by performing O_3 . Components C_1 and C_2 are assembled (operation A_1). This results in component C_3 , which is further transformed to obtain component C_4 . Raw material H_3 leads to component C_5 by performing successively operations O_5 , O_6 and O_7 . Finally, components C_4 and C_5 are assembled through assembly operation A_2 to produce the finished product P .

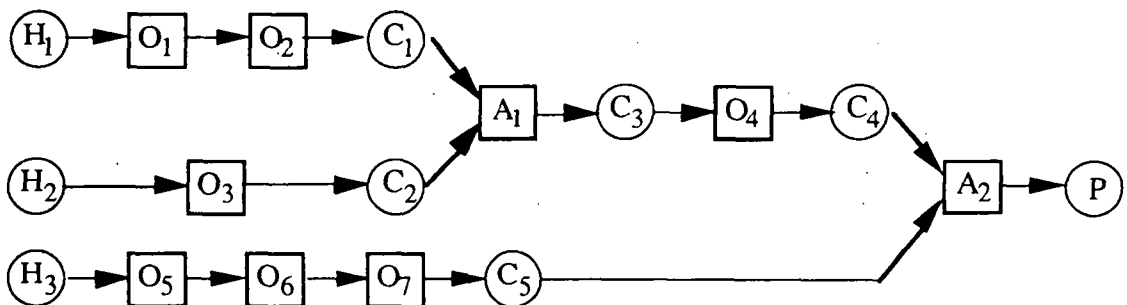


Fig. 1: A manufacturing process

Note that usually several machines can perform a given operation, and that a machine can be used to perform different operations. The time required to perform an operation may be different from one machine to another.

As far as manufacturing systems are concerned, the following constraints hold:

- (i) an operation is completely performed on the same machine, and an operation which starts should be completed without any break, i.e. preemption is not allowed,
- (ii) at most one operation is performed on each machine at a time.

When modeling a manufacturing process, a transition represents a pair (operation, machine), and the firing time of the transition is the time needed to perform the operation on the machine. As a consequence, one operation is represented by as many transitions as the number of machines which can perform this operation. All these transitions have a common input place and common output place which contain respectively the semi-finished product before and after performing the operation. The output places whose output transitions belong to T_0 contain finished products.

A token which represents one unit of raw material is introduced in the system by firing an input transition, while a token which represents a finished product leaves the system by firing an output transition.

To illustrate the modeling of a manufacturing process, let us consider the part of the manufacturing process represented in Figure 1 which starts with raw materials H_1 and H_2 and ends with component C_3 . Let us assume that O_1 can be performed using either machine M_1 or machine M_2 , the manufacturing times being respectively 2 and 3 units of time. Let us also assume that O_2 is only performed on machine M_3 (manufacturing time 4), O_3 either on M_2 (manufacturing time 5) or M_3 (manufacturing time 4), and A_1 either on M_4 (time 1) or M_5 (time 2).

The related model is represented in Figure 2.

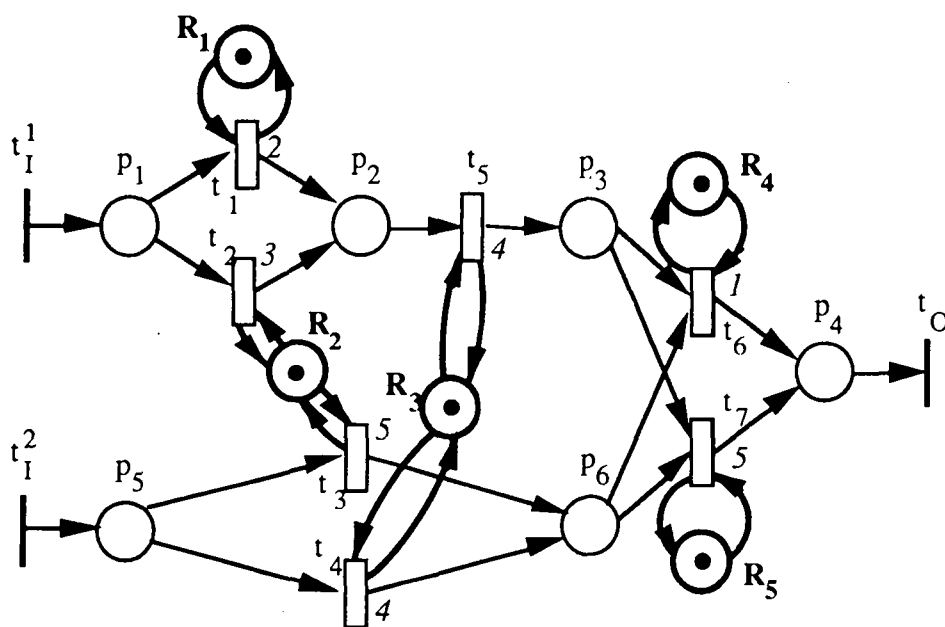


Fig. 2: Partial model of the manufacturing process of Figure 1

In this model, P_1 (resp. P_5) contains tokens which represent the raw material H_1 (resp. H_2). Assuming that \equiv stands here for "represents", we have:

$$t_1 \equiv (M_1, O_1); t_2 \equiv (M_2, O_1); t_3 \equiv (M_2, O_3); t_4 \equiv (M_3, O_3); \\ t_5 \equiv (M_3, O_2); t_6 \equiv (M_4, A_1); t_7 \equiv (M_5, A_1)$$

Tokens in P_3 represent components C_1 while tokens in P_6 represent components C_2 and tokens in P_4 represent components C_3 . Each time t_0 fires corresponds to a unit of C_3 leaving this part of the system.

R_i ($i = 1,2,3,4,5$) are the resource places. Each resource place contains one token: this prevents the machines to be used for manufacturing more than one product at a time. For instance, the firing of t_1 represents operation O_1 performed on M_1 . During the time t_1 is firing, there is no token in R_1 , which means that another firing of t_1 cannot start. When the firing of t_1 ends, a token appears in R_1 and t_1 can start firing again if there is at least one token in P_1 : this means that machine M_1 is idle and that another operation can start on the same machine. Similarly, when t_2 fires, it means that O_1 is performed on M_2 . During the time t_2 is firing, the token in R_2 vanishes, which prevents another firing of t_2 (i.e. the use of M_2 to perform O_1 one more time simultaneously), or a firing of t_3 (i.e. the use of M_2 to perform O_3 while M_2 is already used to perform O_2).

When a resource place R has only one input/output transition t , (R,t) is called a self-loop. When a transition does not share a resource place with another transition, we associate a self-loop to this transition. For the sake of simplicity, we do not represent the self-loops from now on.

Note that the partial model of the manufacturing process is a CO net. We can easily prove that the model of the complete manufacturing process, or the model of a set of manufacturing processes using the same set of machines, is also a CO net, each output transition corresponding to one part type. In such a CO net, a minimal t-invariant is a routing.

If we consider the conditions which hold for a CO net from the point of view of a manufacturing system, we can see that:

- (i) (K1) and (K2) prevent a machine t from performing more than one operation at a time;
- (ii) G' is the set of manufacturing processes (condition (K3));
- (iii) It is possible to introduce raw material in the system (by firing transitions which belong to T_I) and to exit finished products from the system (by firing transitions of T_O). The process places guarantee that any semi-finished product can wait in front of the next machine (condition (K4));
- (iv) Finally, condition (K5) guarantees that it is possible to manufacture a product type without being obliged to manufacture another type of product. Condition (K5) also guarantees that all the available routings can be used.

3. Problem formulation

Let \mathcal{M} be the CO net model of a manufacturing system S . If the number of part types to be manufactured is N , then $|T_O| = N$, where T_O is the set of output transitions of \mathcal{M} .

To illustrate the problem, we will use the model presented in Figure 3 in the remainder of the paper. This model concerns two product types W_1 and W_2 . W_1 is obtained by performing two operations O_1 and O_2 . O_1 can be performed either on machine M_1 or on machine M_2 , while O_2 is performed only on machine M_3 . W_2 results from the assembly of two components C_1 and C_2 . Component C_1 is obtained by performing operation O_3 on M_2 , while C_2 is the result of operation O_4 performed on M_4 . The assembly operation O_5 is performed on M_3 .

We assume that we know, for each product type i , the ratio μ_i to be manufactured during a given period Y . θ_t is the time required to fire transition $t \in T$. We assume that θ_t is deterministic. The goal is to find a schedule which maximizes the throughput of the system and meets the required ratios while keeping the marking as small as possible. From a practical point of view, the goal is to maximize the productivity and meet the required ratios while minimizing the Work-In-Process (WIP).

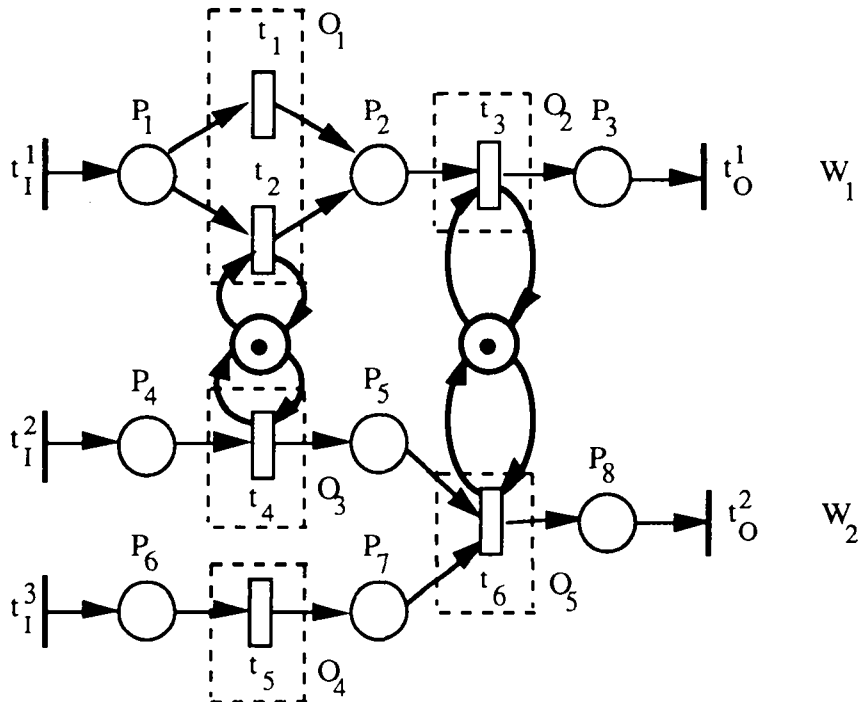


Fig. 3: A CO model for two product types

4. Continuous formulation of the planning problem

If n_t is the number of times $t \in T$ fires during period Y , the planning problem to be solved can be written as follows, and is referred to as \mathcal{P} in the following of this paper:

$$\text{Max } n_{t^*}, \text{ where } t^* \text{ is any of the transitions belonging to } T_O, \quad (1)$$

such that:

$$1 - \forall q \in R, \sum_{t \in q^0} n_t \theta_t \leq Y \quad (2)$$

R includes the places belonging to a self-loop.

$$2 - \forall p \in P, \sum_{t \in p^0} n_t = \sum_{t \in p^0} n_t \quad (3)$$

$$3 - \forall t \in T_I, n_t \leq \lfloor Y \varphi_m(t) \rfloor \quad (4)$$

where $\varphi_m(t)$ is the maximal firing frequency of t , and $\lfloor a \rfloor$ represents the greatest integer less than or equal to a . These frequencies are known for $t \in T_I$.

$$4 - \forall t \in T_O, \frac{n_t}{\sum_{s \in T_O} n_s} \leq \mu_t < \frac{n_t + 1}{\sum_{s \in T_O} n_s + 1} \quad (5)$$

where μ_t is the ratio μ_i if product type i corresponds to output transition t .

(1) indicates that we want to maximize the number of firings of one of the output transitions, which results in maximizing the number of firings of all the output transitions, since the firing of the output transitions verify the given ratios.

Constraints (2) are the capacity constraints. They mean that any machine cannot be busy for a time greater than the manufacturing period Y . θ_t is assumed to be deterministic.

Constraints (3) indicate that if operation O_j follows operation O_s on a product P , then the number of times O_j is performed is equal to the number of times O_s is performed. This results in keeping the buffer levels strictly positive, which is desirable since buffers are supposed to absorb random events.

Inequalities (4) are the constraints on the number of firings of the input transitions. They guarantee that this number can be reached taking into account the flow of raw material arriving in the system.

Finally, inequalities (5) are introduced to make sure that the ratios reached are as close as possible to the desired ratios.

The planning problem \mathcal{P} does not take into account the scheduling of the parts on the machines, but only the load of the machines. As a consequence, the productivity derived from the optimal solution of \mathcal{P} may be not feasible, and is an upper bound of the optimal solution of our scheduling problem.

We consider the continuous formulation of the planning problem, that is the formulation in terms of frequencies. $\varphi(t)$ is the firing frequency of t . The problem is formulated as follows:

$$\mathbf{Max} \varphi(t^*) \text{ for a } t^* \in T_O, \quad (1')$$

such that:

$$1 - \forall q \in R, \sum_{t \in q^0} \varphi(t) \theta_t \leq 1 \quad (2')$$

$$2 - \forall p \in P, \sum_{t \in p^0} \varphi(t) = \sum_{t \in p^0} \varphi(t) \quad (3')$$

$$3 - \forall t \in T_I, \varphi(t) \leq \varphi_m(t) \quad (4')$$

$$4 - \forall t \in T_0, \frac{\varphi(t)}{\sum_{s \in T_0} \varphi(s)} \leq \mu_t < \frac{[\varphi(t) + 1/Y]}{\left[\sum_{s \in T_0} \varphi(s) + 1/Y \right]} \quad (5')$$

This problem is referred to as \mathcal{P}_C in the remainder of this paper.

Note that \mathcal{P} is an integer linear programming problem while \mathcal{P}_C is a real linear programming problem. Thus \mathcal{P}_C , unlike \mathcal{P} , is easy to solve.

The following result connects the optimal solution of \mathcal{P}_C to the optimal solution of \mathcal{P} .

Result 1. *Assuming that $t^* \in T_0$ is the same transition in the formulation of problem \mathcal{P} than in the formulation of problem \mathcal{P}_C , the optimal value of the criterion of problem \mathcal{P}_C multiplied by Y is an upper bound of the optimal value of the criterion of problem \mathcal{P} .*

Proof. Let $\{\tilde{n}_t\}_{t \in T}$ be the optimal solution to problem \mathcal{P} . Then $\{\tilde{n}_t/Y = \varphi(t)\}_{t \in T}$ is a feasible solution to problem \mathcal{P}_C since:

- inequalities (2') hold for $\varphi(t)$ if inequalities (2) hold for \tilde{n}_t since (2') is derived from (2) by dividing both sides of the inequality by Y .
- similarly, inequalities (3') hold for $\varphi(t)$ if inequalities (3) hold for \tilde{n}_t .
- inequalities (4) lead to $\tilde{n}_t \leq Y \cdot \varphi_m(t)$ and, by dividing both sides by Y : $\varphi(t) \leq \varphi_m(t)$: inequalities (4') hold.
- finally, (5') is derived from (5) by dividing the numerators and the denominators of the fractions by Y .

Thus, whatever $t^* \in T_0$:

$$\tilde{n}_{t^*} / Y = \varphi(t^*) \leq \tilde{\varphi}(t^*), \text{ optimal value of the criterion of problem } \mathcal{P}_C.$$

Finally:

$$\tilde{n}_{t^*} \leq Y \cdot \tilde{\varphi}(t^*)$$

Q.E.D.

5. A scheduling for optimal productivity

In this subsection, we show how to take advantage of the optimal solution of the continuous problem \mathcal{P}_C to maximize the productivity of the system. More precisely, we will prove that:

(i) it is possible to reach the flows which are the optimal solutions to problem \mathcal{P}_C in a limited time, and

(ii) it is possible to derive a scheduling process from the previous result in order to reach the optimal flows while keeping the WIP as small as possible.

(i) Reaching the optimal flows of \mathcal{P}_C

Assuming that the $\varphi_m(t)$ are rational numbers, that the ratios μ_t are also rational numbers, and that Y is also rational, we know that there exists an optimal solution $\{\tilde{\varphi}(t)\}_{t \in T}$ whose components are rational. Thus these flows can be written as $\tilde{\varphi}(t) = a_1(t) / a_2(t)$, where $a_1(t)$ and $a_2(t)$ are integer values. Let A be the lowest common multiple (LCM) of the set $\{a_2(t)\}_{t \in T}$ of integer values. Then $\tilde{\varphi}(t) = B(t) / A$ for any $t \in T$, where $B(t)$ is an integer value.

From inequalities (2') we derive:

$$\sum_{t \in q^0} B(t)\theta_t \leq A, \quad \forall q \in R$$

These inequalities show that each $t \in q^0$ can be fire $B(t)$ times during period A without any overlap between firings, assuming that enough tokens are available in each $p \in P \cap {}^0(q^0)$.

From equalities (3'), we obtain:

$$\sum_{t \in {}^0 p} B(t) = \sum_{t \in p^0} B(t), \quad \forall p \in P$$

Thus, the total number of firings of the transitions of p^0 during period A is the same for any p belonging to the model of the same manufacturing process.

For instance, if we refer to Figure 3, the total number of firings of t_1 and t_2 during period A is the same as the number of firings of t_3 during the same period. Likewise, the number of firings of t_4 , t_5 and t_6 during period A are the same, but they are usually different from the number of firings of t_3 . Since one output transition corresponds to one manufacturing process, we refer to this number of firings by $N_A(t)$, $t \in T_O$. We also refer to a place p , which belongs to the same manufacturing process model as $t \in T_O$, as a predecessor of t , and we refer to $t \in T_O$ as a successor of p .

We now assume that initially each $p \in P$ contains $N_A(t)$ tokens, where $t \in T_O$ is a successor of p . Some of these tokens are real tokens, i.e. tokens which compose the initial marking. Place p contains $M_0(p)$ such tokens, where M_0 is the initial marking of the system. The remainder, that is $N_A(t) - M_0(p)$, are artificial tokens. These artificial tokens can fire the transitions exactly as real tokens do.

If $M_0(p) > N_A(t)$, then all the tokens are real, and we consider that $M_0(p) - N_A(t)$ tokens are frozen in place p : they will never be used. Furthermore, since we assume that there is no limit on raw material, and since the firing time of the input transitions is equal to 0, we can assume that they are $N_A(t)$ real tokens in any place $p \in (t^*)^0$, where $t^* \in T_I$ and $t \in T_O$ belongs to the same manufacturing process model as t^* .

We now consider the following constrained operating mode (COM). At the beginning of each period A , we start firing the $N_A(t)$ tokens contained in each place. Transitions fire as soon as they are enabled, but we do not allow a token to fire more than one transition during the same period A . Thus, the $N_A(t)$ real or artificial tokens which are in a place $p = t^0 \in P$ predecessor of $t \in T_I$, at the beginning of a period A , will be available in $(p^0)^0 \cap P$ at the beginning of the next period A , and

so on. As a consequence, the set of tokens which fire an output transition are composed only with real tokens at the latest after a time $K.A$, where K is the maximal number of places minus 1 on the paths joining the input transitions to the output transition under consideration, each path being included in only one manufacturing process model. We assume that the output transitions, whose firing times are equal to zero, fire as soon as they are enabled.

For instance, if we consider Figure 3, both t_O^1 and t_O^2 will be fired with real tokens at the latest after a time $2.A$. This means that, under the constrained operating mode previously described, the system is in a steady state which meets the optimal flows solution of problem \mathcal{P}_C at the latest at time $2.A$. This, of course, remains if transitions fire as soon as they are enabled and if we relax the COM.

Thus, result 2 holds.

Result 2. *If transitions fire as soon as they are enabled, the system is in a steady state which meet the optimal flows solution of problem \mathcal{P}_C at the latest at time $K.A$ where:*

- K is equal to $\text{Max}_{t \in T_O} (Z_t - 1)$, where Z_t is the number of places belonging to the manufacturing process model corresponding to $t \in T_O$,
- A has been previously defined.

Since the optimal value of the criterion of \mathcal{P}_C provides an upper bound of the optimal value of the criterion of \mathcal{P} , since the optimal value of the criterion of \mathcal{P} gives an upper bound of the optimal criterion of the scheduling problem, and since we exhibit a schedule whose criterion value reaches the upper bound provided by the optimal solution of \mathcal{P}_C , we can claim that this schedule is optimal.

Result 3. *The previous schedule is optimal at the latest from time $K.A$ onwards.*

(ii) A scheduling process

From the previous subsection, it turns out that, if each $p \in P$ contains initially $N_A(t)$ tokens, where $t \in T_O$ is the successor of p , then the system is in steady state from time 0 onwards. Indeed, this statement also holds if we relax the COM by allowing a token to fire a transition more than once during the same period A .

The first goal is to minimize the total number of tokens which are initially in the places of P while keeping the system in steady state from time 0 onwards. We will then show how to reach these state whatever the initial marking is.

We first propose an heuristic algorithm to minimize the total number of tokens.

a. Minimizing the total number of tokens

Assume that initially $M_0(p) = N_A(t)$, for any $p \in P$, where $t \in T_O$ is the successor of p .

We simulate the system following the rule presented hereafter:

- when an event arises at time τ , that is when a token appears in a resource place r , we consider the subset T_r^* of r^0 such that, if $t \in T_r^*$, the number $n_t(\tau)$ of firing starts of transition t until time τ is less than $\bar{\varphi}(t).A$.
- if $T_r^* = \emptyset$, no action is taken, and we wait for the next event.

otherwise, we compute:

(6)

$$s_t(\tau) = \bar{\varphi}(t).\tau - n_t(\tau) \text{ for each } t \in T_r^* \text{ and we start firing } t_a \in T_r^* \text{ such that:}$$

$$s_{t_a}(\tau) = \text{Max}_{t \in T_r^*} s_t(\tau)$$

Note that, if $T_r^* \neq \emptyset$, then $s_{t_a}(t) > 0$.

We perform the simulation on a period A and keep, for each $p \in P$, the minimal marking $M_m(p)$ during this period.

Thus, if $M_0(p) = N_A(t) - M_m(p)$, for any $p \in P$, the system is in steady state from time 0 onwards and the productivity is maximal, assuming that the above scheduling rule (6) holds. We refer to this marking as the reduced initial marking.

b. Controlling the transient state

In the previous subsection, we defined a reduced initial marking which guarantees that the system is in steady state as soon as the system starts running, assuming that a idle transition starts firing as soon as it is enabled.

Unfortunately, due to machine breakdowns and/or the discrepancy between the scheduling decisions and their application and/or the changes in the production ratios, it may happen that the initial marking \bar{M}_0 is different from the reduced initial marking M_0 related to the production ratios to be met. In this case, we have to control the transient state which will move the marking from \bar{M}_0 to M_0 .

Let \bar{n}_t be the number of firings which should be performed on $t \in T$ to reach the marking M_0 .

The following relations hold:

$$\sum_{t \in p^0} \bar{n}_t - \sum_{t \in^0 p} \bar{n}_t = \bar{M}_0(p) - M_0(p), \quad \text{for any } p \in P \quad (7)$$

Let us consider $t_0 \in T_0$ and $\bar{p} \in P$ such that there exists a directed path from \bar{p} to t_0 which does not contain a resource place (\bar{p} is essor of t_0). We denote by $P(\bar{p}, t_0) \subset P$ the set of places which belong to this path. Then, by adding side by side relationships (7) for all the $p \in P(\bar{p}, t_0)$, we obtain:

$$\sum_{s \in^0 \bar{p}} \bar{n}_s = \bar{n}_{t_0} + \sum_{p \in P(\bar{p}, t_0)} (M_0(p) - \bar{M}_0(p)) \quad (8)$$

Relationship (8) holds for any $\bar{p} \in P$ and $t_0 \in T_0$, where \bar{p} is a predecessor of t_0 . The number of relationships (8) is $|P|$. Since we want to meet the production ratios, $\bar{n}_{t_0} = K.B(t_0)$ for any $t_0 \in T_0$, where K is the smallest positive integer such that:

$$K \geq \text{Max}_{(\bar{p}, t_0)} \left[- \sum_{p \in P(\bar{p}, t_0)} (M_0(p) - \bar{M}_0(p)) / B(t_0) \right] \quad (9)$$

Result 4. For any pair (\bar{p}, t_0) , $\bar{p} \in P$, $t_0 \in T_0$, where \bar{p} is a predecessor of t_0 , the number of firings of the transitions of ${}^o\bar{p}$ required to reach the steady state is given by:

$$\sum_{s \in {}^o\bar{p}} \bar{n}_s = K \cdot B(t_0) + \sum_{p \in P(\bar{p}, t_0)} (M_0(p) - \bar{M}_0(p))$$

where K is the smallest positive integer which verifies (9).

Consequently, we can reach the steady state by firing the transitions of ${}^o\bar{p}$, $\bar{p} \in P$, as soon as they are enabled, until $\sum_{s \in {}^o\bar{p}} \bar{n}_s$ firings are performed. If a conflict arise, the transition to be fired is

chosen at random. The steady state starts when, for any $\bar{p} \in P$, the transitions of ${}^o\bar{p}$ have been fired $\sum_{s \in {}^o\bar{p}} \bar{n}_s$ times.

c. The scheduling algorithm

The scheduling algorithm has three main steps:

1. We first apply the heuristic algorithm provided in subsection 5.a. in order to define the so called reduced marking.
2. We then apply result 4 in order to reach the steady state.
3. We then continue to fire the transitions on a EOM (Earliest Operating Mode) basis ; in other words, we fire the transitions as soon as they are enabled.

The algorithm stops as soon as either customers' requirements, or the horizon Y , are reached.

6. Some numerical examples

We consider the model given in Figure 4. In this figure, the resource places are not represented for simplicity. This model represents a system of 6 machines performing 5 types of products. The machines related to each transition, as well as the corresponding firing times, are given in the figure.

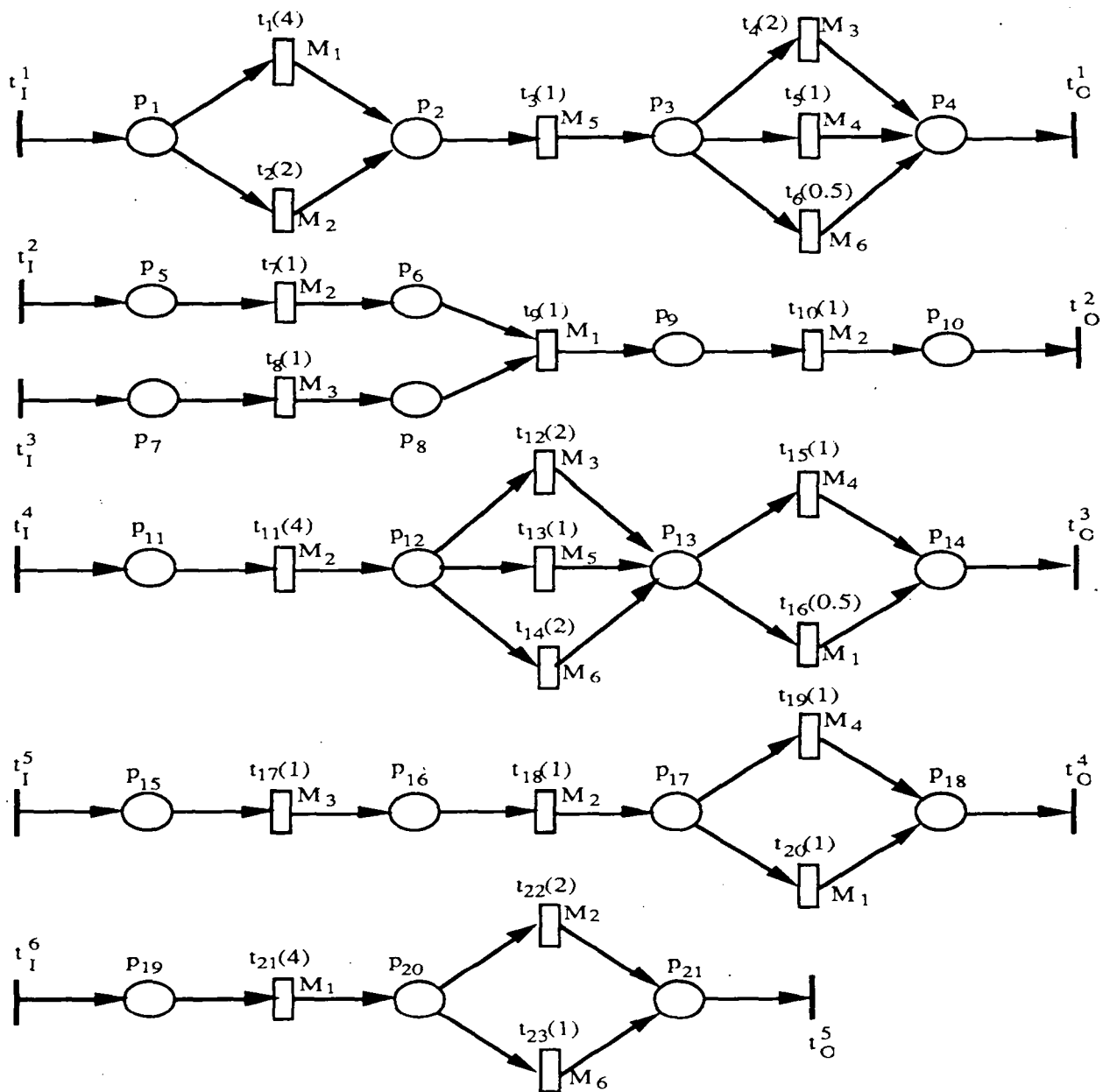


Fig. 4: An example

6.1. First example

In this run, the ratios are:

$$\langle 0.2 ; 0.2 ; 0.2 ; 0.2 ; 0.2 \rangle$$

The initial marking is:

$$\langle 0, 3, 1, 0, 0, 2, 0, 0, 0, 0, 0, 2, 1, 0, 0, 2, 0, 0, 0, 2, 0 \rangle$$

The computation leads to the following reduced marking:

$$\langle 0, 2, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 2, 0, 0, 0, 3, 0, 0, 3, 0 \rangle$$

The time needed to reach the steady state is $\tau = 24$.

Periodicity in steady state: $A = 23$.

We consider the output transition t_O^1 . The firing frequency is:

in transient state: $3 / \tau = 0.125$

in steady state: $3 / A = 0.130$

6.2. Extension of the previous example

We keep the same ratios as in the previous example, but we modify the initial marking. The results are given in table 1.

Table 1: Examples

| Ex. | Initial marking | τ | Frequency in transient state | Frequency in steady state |
|-----|--|--------|------------------------------|---------------------------|
| 1 | $\langle 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 \rangle$ | 18 | $0 / \tau = 0$ | 0.130 |
| 2 | $\langle 0,2,1,0,0,0,0,0,1,0,0,2,1,0,0,2,2,0,0,0,0 \rangle$ | 31 | $3 / \tau = 0.097$ | 0.130 |
| 3 | $\langle 0,12,11,0,0,0,0,0,1,0,0,15,1,0,0,13,2,0,10,0,0 \rangle$ | 117 | $21 / \tau = 0.179$ | 0.130 |
| 4 | $\langle 0,15,1,0,0,2,0,0,0,0,0,2,1,0,0,2,0,0,0,2,0 \rangle$ | 107 | $15 / \tau = 0.140$ | 0.130 |
| 5 | $\langle 0,2,0,0,0,0,0,0,1,0,0,1,2,0,0,0,3,0,0,3,0 \rangle$ | 0 | $0 / \tau = 0$ | 0.130 |

Time τ is the time needed to reach the steady state, i.e. the state defined by the reduced initial marking. But the firing frequencies, which reflect the productivity of the system, can be higher on $[0, \tau]$ than in the steady state: it is the case when the initial marking is very high compared to the reduced marking (see example 3 for instance). But, if the initial marking is low compared to the reduced marking, the firing frequency may be very low, if not zero, compared to the frequency in steady state (see example 1).

7. Conclusion

In this paper, we propose a scheduling algorithm which:

- (i) defines a reduced marking which allows to maximize the firing frequency by firing the transitions as soon as they are enabled ;
- (ii) moves the initial marking to the reduced marking (transient state) ;
- (ii) continues the firing process by firing the transitions as soon as they are idle, which is possible since the reduced marking guarantees that an idle transition is also enabled.

Further research will focus on the study of an algorithm which will guarantee that the reduced marking is minimal. This should be possible by adjusting the starts of the transition firing at the expense of an additional computation complexity.

Other investigations should be conducted to see if the strategy which consists of targeting the steady state, and then of continuing on an EOM basis, is the one which leads to the best global frequency.

References

- [1] COMMONER F., HOLT A., EVEN S. and PNUELI A., "Marked Directed Graphs", *Journal of Computer and System Science* 5/5 (1971) 511-523.
- [2] DI CESARE F., HARHALAKIS G., PROTH J.M. and SILVA M., "*Practice of Petri Nets in Manufacturing*", Chapman and Hall, 1993
- [3] HILLION H.P. and PROTH J.M., "Performance Evaluation of Job-shop Systems using Timed Event Graphs", *IEEE Transactions on Automatic Control* 34/1 (1989) 3-9.
- [4] LAFTIT S., PROTH J.M. and XIE X.L., "Optimization of Invariant Criteria for Event Graphs", *IEEE Transactions on Automatic Control* 37/5 (1992) 547-555.
- [5] LEVENTOPOULOS M., "A New Class of Petri Nets for Modeling, Planing and Scheduling of Flexible Manufacturing Systems", Thesis Report, Master's Degree, M.S. 94-10, University of Maryland at College Park.
- [6] MURATA T., "Petri Nets: Properties, Analysis and Applications", *Proceedings of the IEEE* 77/4 (1982) 541-580.
- [7] PROTH J.M. and XIE X.L., "Cycle Time of Stochastic Event Graphs: Evaluation and Marking Optimization", *IEEE Transactions on Automatic Control* 39/7 (1994) 1482-1486.
- [8] PROTH J.M., SAUER N., WARDI Y. and XIE X.L., "Marking Optimization of Stochastic Timed Event Graphs using IPA", *Proc. of 32nd IEEE Conference on Decision and Control*, p. 686-691, San Antonio, Texas, December 1993.
- [9] PROTH J.M., SAUER N. and XIE X.L., "Optimization of the Number of Transportation Devices in a Flexible Manufacturing System Using Event Graphs", *Research Report INRIA*, no 2359, 1994.
- [10] PROTH J.M., WANG L. and XIE X.L., "A Class of Petri Nets for Manufacturing Systems Integration", *Research Report INRIA*, no 2055, 1993.
- [11] RAMAMOORTHY C.V. and HO G.S., "Performance Evaluation of Asynchronous Concurrent Systems Using Petri Nets", *IEEE Transactions on Software Engineering (SE)* 6/5 (1980) 440-449.
- [12] SILVA M. and VALETTE R., "Petri Nets and Flexible Manufacturing", in *Advances in Petri Nets 1989*, G. Rozenberg (ed.), Lecture Notes of Computer Science 424, Springer-Verlag, 1989, 374-417.
- [13] SURI R., "Perturbation Analysis: The State of the Art and Research Issues Explained via the G1/G/1 Queue", *Proceedings of the IEEE* 77/1 (1989) 114-137.



Unité de recherche INRIA Lorraine
Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)

Unité de recherche INRIA Rennes - IRISA, Campus universitaire de Beaulieu 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes - 46, avenue Félix Viallet - 38031 Grenoble Cedex 1 (France)
Unité de recherche INRIA Rocquencourt - Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis - 2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)

ISSN 0249 - 6399

