Journal of Intelligent Manufacturing, 16, 67–92, 2005

Modular Petri Net based modeling, analysis, synthesis and performance evaluation of random topology dedicated production systems

G.J. TSINARAKIS and N.C. TSOURVELOUDIS*

Intelligent Systems and Robotics Laboratory, Department of Production Engineering and Management, Technical University of Crete, 73100 Chania, Crete, Greece
E-mail: (tsinar, nikost)@dpem.tuc.gr

K.P. VALAVANIS

Center for Robot-Assisted Search and Rescue, Department of Computer Science and Engineering, University of South Florida, 4201 E. Fowler Avenue, Tampa, FL, 33620-5399, USA
E-mail: kvalan@csee.usf.edu

Received January 2003 and Accepted April 2004

Ordinary *t-timed* modular Petri Nets are used for modeling, analysis, synthesis and performance evaluation of random topology dedicated production systems. Each system is first decomposed into *production line*, *assembly*, *disassembly* and *parallel machine* modules followed by derivation of their modular Petri Net models. Two sets of modules, *generic* and *generalized* respectively, are derived corresponding to the simplest and most general cases. Overall system Petri Net model is obtained via synthesis of the individual modules satisfying system features (production rates, buffer capacities, machine expected up, down or idle times). Detailed mathematical expressions are derived for modules *P-invariants* (and *T-invariants* when exist); they are further generalized for a random topology and complexity dedicated production system. Total number of the individual Petri Net module nodes as well as of the combined system Petri Net model is also calculated. Results show the applicability of the proposed methodology and justify its modeling power and generality.

Keywords: Dedicated production systems, production line, assembly, disassembly, parallel machines, modular Petri Nets, P-invariants, nodes complexity

Nomenclature

Symbol	Explanation
PN	Petri Net
WIP	Work-in-process
DEDS	Discrete event dynamic systems
MRP	Material Requirements Planning
P	Set of places
T	Set of transitions
I	Set of input arcs (to transitions)
O	Set of output arcs (to transitions)

*Author for correspondence.

V	Set of vertices
N	Set of non-negative integers
w_{ij}	Arc weight from node i to node j
m_0	Initial marking
m_i	Marking i
m_f	Final marking
$m_j\ (p_i)$	Number of tokens in place p_i in marking j
$m_0\ (p_i)$	Number of tokens in place p_i with initial marking m_0
$R(m_i),R(m_0)$	Reachability set of marking m_i, marking m_0
n_p	Overall number of places in a Petri Net
n_t	Overall number of transitions in a Petri Net
$A=[a_{ij}]$	Incidence matrix
$r\ =rank\ (A)$	Rank of matrix A
$\alpha_{ij}{}^{+}$, $\alpha_{ij}{}^{-}$	Tokens added, removed, in place j by firing transition i
S	Transition firing sequence
n_{TCi}	Number of machines in generalized transfer chain module $i(0<i<n_1)$
n_{Ai}	Number of input buffers in generalized assembly module $i(0<i<n_2)$
n_{Di}	Number of output buffers in generalized disassembly module $i(0<i<n_3)$
n_{Pi}	Number of parallel machines in generalized parallel machines module $i(0<i<n_4)$
B_i, B_{ji}, B_{il}	Buffer i, upstream buffer, downstream buffer, respectively
b_{ji}, b_{il}	Level of upstream, downstream buffer, respectively
UL_i	Upper limit of token number found concurrently in any place of module i, where i is *TC* for transfer chain module, *A* for assembly module, *D* for disassembly module and *P* for parallel machines
M_j	Machine j
C_i	Capacity of buffer i
n_1	Number of transfer chain modules in the overall system
n_2	Number of assembly modules in the overall system
n_3	Number of disassembly modules in the overall system
n_4	Number of parallel machine modules in the overall system
n_{E}	External input places
n_{F}	Final buffers of the net (output places of the net)
$p_{i\text{-}j}$	Fused place created from the fusion of places p_i and p_j
k_j	Number of parts preserved in *P-invariant i*
X	$(n_p\ x\ 1)$ non-negative integer vector
Y	$(n_t\ x\ 1)$ non-negative integer vector
r_i	Processing rate
s_i	State of machine M_i
d_1	Number of machines of a production system
d_2	Number of *P*-invariants of the total system referring to mutually exclusive states of the machines

1. Introduction

In this paper, a concrete step-by-step methodology is presented using *t-timed ordinary modular PNs* for modeling, analysis, synthesis, performance evaluation and simulation of random topology dedicated production systems, viewed as surplus based systems, composed of a network of workstations and buffers, where machines fail and are repaired randomly. Petri Nets (PNs) and their extensions being both a mathematical and graphical tool are widely used for modeling discrete event dynamic systems (DEDS) including production systems[1] and networks (Valavanis, 1990; Zhou and DiCesare, 1993; Desrochers and Al-Jaar, 1995; Proth and Xiaolan, 1996; Moody and Antsaklis, 1998). PNs have been proven to be a powerful tool for studying system concurrency, sequential, parallel, asynchronous, distributed deterministic or stochastic behavior, resource allocation, mutual exclusion and conflicts (Murata, 1989; Desrochers and Al-Jaar, 1995; Valette *et al.*, 1999; Gu and Bahri, 2002).

This work is the natural outgrowth of previously published (Tsourveloudis *et al.*, 2000; Ioannidis *et al.*, 2002) research that studied single/multiple-part-type and re-entrant manufacturing networks (systems) of random topology and complexity where random machine breakdowns and repairs take place. In these works, a decomposition method is used for scheduling of manufacturing systems through minimizing work-in-process (WIP) and cycle time while maintaining at the same time quality of service by keeping backlog at acceptable levels. This is accomplished by deriving a set of distributed single-level fuzzy controllers to monitor production rates locally and a supervisory fuzzy logic controller to monitor overall system production rate. Except manufacturing this methodology can be used in supply chain in the framework of logistics.

For a random topology dedicated production system or network[2], whether single- or multiple-part-type or cyclically scheduled, with finite or infinite capacity buffers, the following are accomplished:

- *Production system decomposition – analysis*: The random topology production system is decomposed into four sets of modules, corresponding to the *production line* or *transfer chain*,[3] *assembly*, *disassembly* and *parallel machines* modules. Their respective modular PN models are derived. The proposed system component modules and their respective PN equivalents refer to the simplest and most general scenario:

  (i) *Generic modules* and *generic PN modules* that correspond to the one input buffer-one machine-one output buffer transfer chain module, two input buffers-one machine-one output buffer assembly module, one input buffer-one machine-two output buffers disassembly module and two parallel machines module;

  (ii) *Generalized modules* and *generalized PN modules* that correspond to the n_{TCi} machines – (n_{TCi} +1) buffers transfer chain module, n_{Ai} input buffers – one machine – one output buffer assembly module, one input buffer – one machine – n_{Di} output buffers disassembly module and one input buffer – n_{Pi} parallel machines – one output buffer module.

- *Production system composition – synthesis*: The overall production system PN model is obtained via synthesis of the component PN models considering component connectivity and complexity as follows:

  (i) *Component connectivity*: This is determined by *places fusion* at the respective module connecting points. (For example considering two sequential transfer chains, the output buffer of the first and the input buffer of the second are fused to one place, connecting the two modules.)

  (ii) *Component complexity and overall system complexity*: This is determined by calculating for any random topology dedicated production system, the total number of the PN module nodes (places, transitions) and the overall PN system nodes.

- *Production system constraints*: The Martinez and Silva algorithm (1982) is used to calculate the *PN* module as well as the overall PN system model *P*- and *T*-invariants, considering machine and buffer operational constraints (capacities, production rates). Mathematical expressions are derived for the basic or minimal PN module invariants, including the invariants that result from module synthesis.
- *Production system simulation and performance evaluation*: The production system is simulated and evaluated through the corresponding PN model. As such, buffer levels, machine utilization (up-time down-time, idle-time), production rates, cycle times and overall production time are calculated and if necessary modified based on specifications and constraints.

The major contributions of the proposed methodology may be summarized as follows: (i) the modular PN based system modeling, analysis, synthesis and performance evaluation is independent of the system architecture and structure, (ii) the model construction method may be extended and slightly modified in order to be applied to any configuration DEDS, (iii) analysis and synthesis of any complicated system is accomplished in terms of analysis and synthesis of the generalized PN modules, which means that the overall complexity is significantly reduced, (iv) there is no limitation on the production system number of machines/workstations and the structure of the net, while buffer capacities may be considered either finite or infinite, (v) system's complexity is easily obtained since the calculation of the overall PN system nodes and invariants are done in general (theoretically) without considering a specific topology system, (vi) whole system's analysis and properties are obtained with respect to the corresponding characteristics of the fundamental modules, (vii) the generic models are simple, based in realistic assumptions and are easily applied and understandable.

While reviewing related literature, several differences have been observed. In Zuberek and Kubiak (1999) Timed PNs are used to model and study manufacturing cells. The possible schedules and routes according to machine number are studied. Each machine is represented by a transition with an input and output arc without considering machine states. Net invariants and reductions are used for performance evaluation, while no behavioral analysis is performed. In Allam and Alla (1998) Hybrid PNs are used for modeling and simulation of an assembly. Machines perform different operations, most of which are modeled using a general model with minor changes, according to specific features. Machines are considered either operational or idle (no breakdown). In Proth and Xiaolan (1996) common manufacturing system PN models are presented (e.g. machines prone to failure, assembly). These models are quite different from the ones presented in this paper in assumptions, structural characteristics (empty machine's signal in not combined with breakdowns, machine breakdowns considered only in a model) and properties. No analysis is curried out, nor simulation and performance evaluation is implemented. In Valavanis (1990) Extended PNs are used to model FMS's in a 3-phase hierarchical procedure (top-down system's decomposition, component net modeling and system's EPN synthesis). Three basic models are used, corresponding to machining station, robot and buffer. Behavior analysis is partial and does not rely on module analysis. In Proth and Sauer (1998) controllable output nets are used to optimize scheduling and maximize the productivity of a job shop. Operations in machines and assemblies transform raw materials in final products with piecewise constant flows. Machines perform multiple operations, each represented by a transition. In Thomas *et al.* (1996) a class of TPNs is used as a representation and analysis framework for automated assembly. The objective is to define the optimal assembly sequence, which can generate a control plan with minimum costs. For this, performance and throughput evaluation is used. In Tang *et al.* (2001) PNs are used for demanufacturing, scheduling and disassembly in order to derive the paths with maximal end-of-life value. PNs model flexible disassembly sequences, machine scheduling and workstation status. No analysis or presentation of analytical models is done. In Jehng (2002) PNs are used for modeling, performance analysis and simulation of automated pressing systems. PN reduction is used. Basic model properties are calculated and zero inventories are investigated to estimate what reduction of setup times allows the adoption of JIT production models. In Xue *et al.* (1998) generalized stochastic PNs model FMSs. Each model is partitioned into a physical subnet (consisting of three transportation resources, cell and job shop subnet that generates transfer and processing parts requests) and a controller subnet. In Wang (1996) and Wang and Wu (1998), Object Oriented and Colored Timed Object Oriented PNs are used for modeling and analysis of Automated Manufacturing Systems. In Wang (1996), physical object classes are identified, modeled and connected to represent AMS control logic. After the implementation of the basic model, the complete OPN model is constructed by adding constraints and attributes of the system. The model is analysed for deadlock detection and invariants. In (Wang and Wu (1998), in the initial method time is added in a 4th phase where CTOPN model is constructed by the OPN model. Then a search algorithm generates a near optimal schedule. System's nodes complexity is high to model even simple systems.

Further, as opposed to several other PN based manufacturing systems modeling approaches

(Peterson, 1981; Valavanis, 1990; Reisig, 1992; Wang and Xie, 1996; Veltsos, 1998; Zuberek, 1998; Balduzzi *et al.*, 2000; Christensen and Petrucci, 2000; Frey, 2000; Domg and Chen, 2001; Jeng, 2001; Jehng, 2002; Tsinarakis, 2002) and references therein, the proposed method is rather complete and general covering the aspects of system decomposition/composition, constraint satisfaction, complexity and properties analysis, performance evaluation and optimization with minimum assumptions, regardless of system topology.

As previously mentioned, the proposed approach is based on the scheduling policy that was suggested by Tsourveloudis *et al.* (2000) and Ioannidis *et al.* (2002). In these works, fuzzy logic was the mathematical tool that used to model generic manufacturing environments. Here, instead of fuzzy logic, we use modular PNs to study realistic manufacturing systems. In addition, we have extended the ideas used for effective modeling to more complex situations like parallel working machines. Nevertheless a significant advantage, observed while evaluating the results, is that although the modular PN based method is drastically different to the fuzzy logic controller, the results are remarkably complementary. For example, minimization of WIP leads to certain (input, internal and output) buffer levels, a phenomenon that is also captured by the derived modular PN system model.[4] The same holds when considering machine blocking and/or starvation. This postulated observation is an additional justification of the generality and applicability of the proposed method.

The rest of the paper is organized as follows: Section 2 presents the basic production system modules and summarizes functionality issues. Section 3 derives their respective PN modules and Section 4 presents the overall system PN model as synthesis of the four corresponding (sets of) modules. Section 5 presents simulation results, while Section 6 concludes the paper.

2. Production system modules

2.1. *Background*

A production system is usually viewed as a network of machines/workstations and buffers. Items receive an operation at each machine and wait for the next operation in a buffer with finite capacity. Random machine breakdowns disturb the production process and phenomena such as starvation

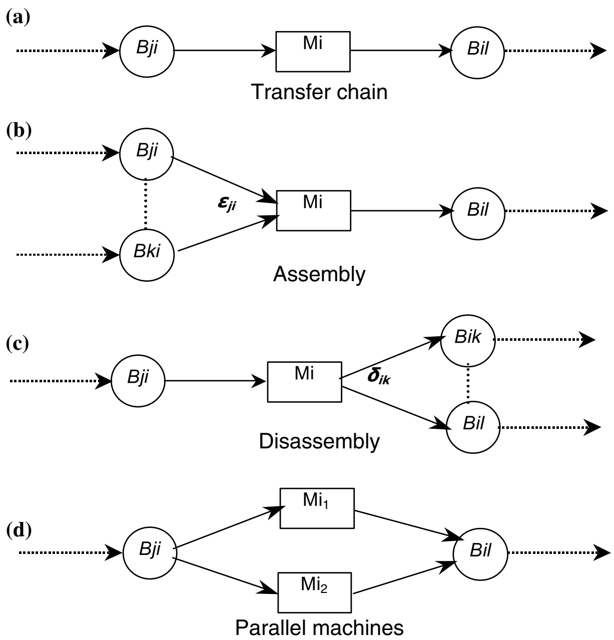

Fig. 1. Generic (fundamental) production system modules.

and/or blocking, may occur. Due to a failed machine with operational neighbors, the level of the downstream buffer decreases, while the level of the upstream buffer increases. If the repair time is large enough, the broken machine will either block the next station or starve the previous one. This adverse effect will propagate throughout the system.

Production control policies may be classified as token-based, time-based and surplus-based (Conway *et al.* 1988; Bai, 1990, 1994; Buzacott and Gershwin, 1993; Custodio *et al.* 1994; Gershwin, 1994; Gershwin, 2000; Tsourveloudis *et al.* 2000; Ioannidis *et al.* 2002). Token-based systems involve token movement in the manufacturing system to trigger events. Time-based systems operate on a time basis; for example, Material Requirements Planning (MRP) systems attempt to determine the time at which an operation should take place. In surplus-based systems, decisions (hedging-point, two-boundary and base-stock policies) are made on the basis of how far cumulative production is ahead of or behind cumulative demand. Surplus-based systems may be viewed as a consequence of the observation that the most important factor in determining a factory performance is the process by which material is released into the system; the best policy views all or many points as release points to the rest of the factory (Wein, 1988).

When considering simple manufacturing systems, analytical results produced thus far have demonstrated the superiority of surplus-based systems (Akella and Kumar, 1986). More specifically, hedging point policies have been proven optimal in minimizing production cost in single-stage, single-part-type system scheduling. Generalizations to more than one-part-type or production stages have proven to be difficult, but obtained solutions may be successfully applied in real manufacturing systems.

2.2. *Dedicated and modular productions systems*

The production systems concerned here are dedicated machine production systems. In these, each

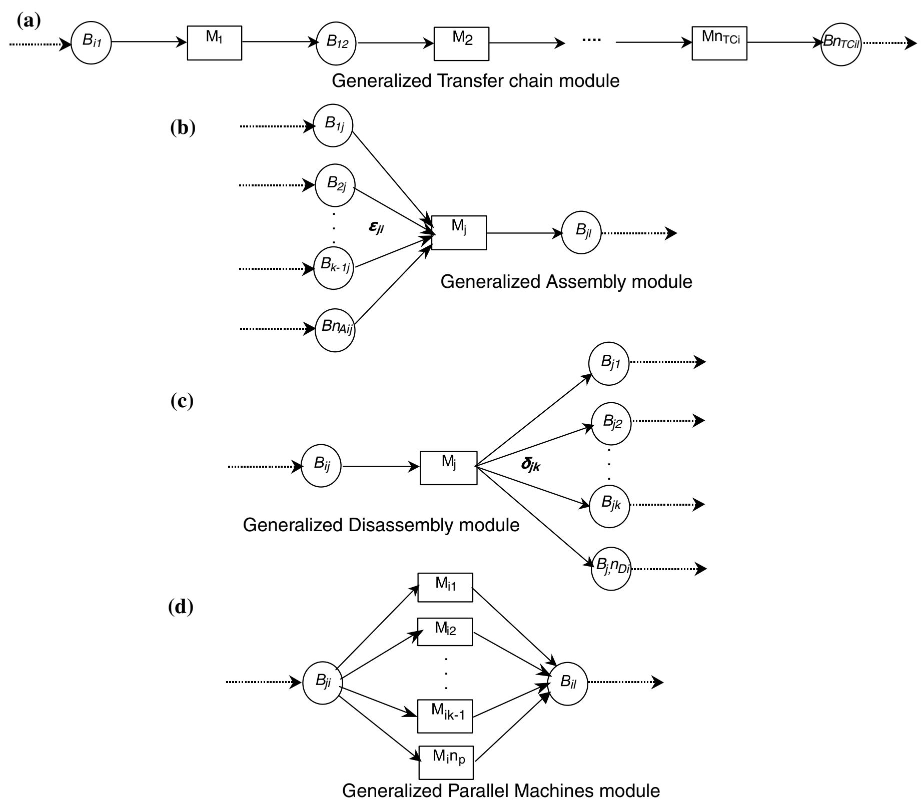

Fig. 2. Generalized production system modules.

machine is assigned exclusively to an operation of one product type, which usually has high demand rate (Proth and Xiaolan, 1996). In such systems all parts follow the same route (pass from the same machines with the same order, their processing times in machines are of the same type and they are obtained from the same final buffer).

According to the proposed approach, it is possible to recognize a small number of subsystems, called from now on fundamental modules. Their corresponding Timed PN models are implemented, analyzed and used as structural components for the representation of the majority of dedicated production systems. Modules are repeated and appropriately connected according to system's topology in order to produce the total model of the system under review.

By using a modular approach, it is possible to simplify the description, qualitative (properties and invariants) and quantitative (mean production cycle, WIP etc) analysis of a complicated system by splitting its complexity into small and simple entities (that can also be considered and analyzed independently each other (Thevenon and Flaur, 2000) and to overcome problems relative to the systems behavior (e.g., some conflicts types), (Feldmann *et al.* 1996). After that, integration of the subsystems must be done in a consistent with the reality way in order to produce the corresponding results for the total system.

The use of modular subsystems in production systems modeling is a need, as this allows the independent modification of the model, results in increased flexibility (required changes to produce a new product are minimized) that meets one of the major requirements of such systems and allows the use of more efficient advanced performance evaluation and analysis techniques, as distributed simulation (Nketsa and Valette, 2001). Moreover modular design of production systems resists obsolescence, shortens redesign, reduces costs, and eases maintenance (Rogers and Bottaci, 1997).

2.3. *Generic and generalized production systems modules*

The production floor modeling approach introduced in Tsourveloudis *et al.* (2000), Ioannidis et al. (2002) is extended so that every production network is decomposed into four *generic modules*: the production line, assembly, disassembly and parallel machines module, the simplest (generic) version of which is shown in Fig. 1 (circles and rectangles represent buffers and machines, respectively). In system's operational phase the use of transferring facilities and similar equipment is not analytically presented.[5]

The line module includes a machine M_i, which takes unfinished items from an upstream buffer B_{ji} and after processing sends them to a downstream buffer B_{il}. In the assembly operation a machine M_i obtains parts from two upstream buffer B_{ji} to B_{ki}, brings them to form a single unit and sends it to a downstream buffer B_{il}. In disassembly a machine M_i takes unfinished parts from an upstream buffer B_{ji}, separates them to two parts and sends them to downstream buffers B_{ik} to B_{il}. The two parallel machines module shows machines with potentially different settings and functions working on the same product. The two machines work independently each other and process parts of the same type. They correspond to the case of two servers for the same customer type. These four modules, if connected to each other may represent manufacturing networks of various layouts.

Generalizations of the generic modules correspond to: n_{TCi} machines- $(n_{TCi}\ +1)$ buffers transfer chain module, n_{Ai} input buffers-one machine-one output buffer assembly, one input buffer-one machine-n_{Di} output buffers disassembly and one input buffer-n_{Pi} parallel machines-one output buffer module as shown in Fig. 2.

Each module may be implemented, tested, evaluated and simulated considering as input variables the buffer levels b_{ji} and b_{ik} of the upstream and downstream buffers and the state s_i of machine M_i.

A buffer tends to be empty when the upstream machine is either under repair or producing at a slower rate than the downstream machine. Similarly a buffer tends to fill when the downstream machine is either under repair or producing at a slower rate than the upstream machine. The information needed to synchronize the operation of the production network is transferred to each control module by the level change of each buffer. Every event occurring in the production network is affecting level of buffers close to the area of the event.

The time that a part spends in a manufacturing system between the start of first operation in a machine and completion of its last operation (time in which process of raw materials starts, until final products are ready) is known as the manufacturing cycle time. Reduction of manufacturing cycle time has many benefits, including lower inventory, reduced costs, faster response to needs and increased flexibility.

3. Petri Net modules

3.1. *Petri Net background and fundamentals*

An Ordinary Petri Net (OPN) is a bipartite directed graph defined as the five-tuple: $PN = \{P, T, I, O, m_0\}$, where $P = \{p_1, p_2 \ldots p_{np}\}$ is a finite set of places, $T = t_1, t_2, \ldots, t_{nt}\}$ is a finite set of transitions, $P \cup T = V$, where V is the set of vertices and $P \cap T = \emptyset$. I: $(P\ x\ T) \rightarrow N$ is an input function and O: $(P\ x\ T) \rightarrow N$ an output function with N a set of non-negative integers, and m_0 the PN initial marking. Places represent conditions; transitions represent events and arcs direct connection, access rights or logical connection between places and transitions.

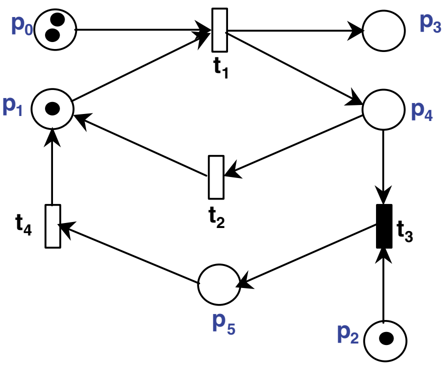

Fig. 3. Production line (chain) generic PN module.

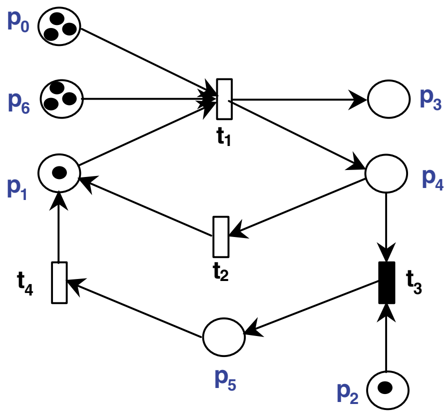

Fig. 4. Assembly generic PN module.

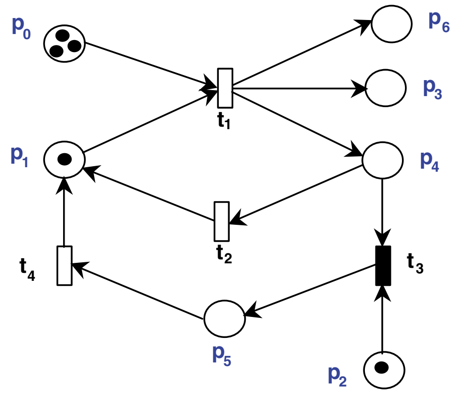

Fig. 5. Disassembly generic PN module.

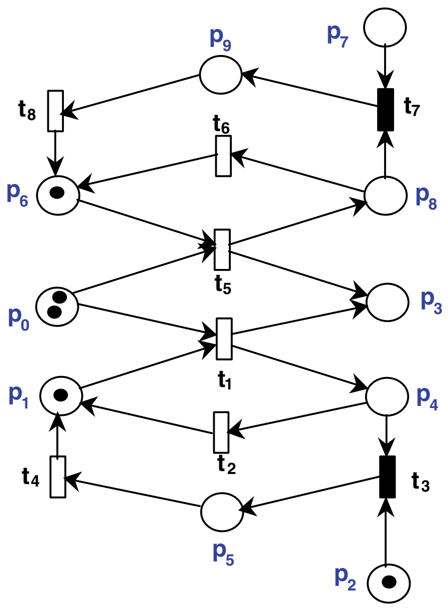

Fig. 6. Two parallel machines generic PN module.

PN structural and behavioral properties (*reachability*, *coverability*, *safeness*, *k-boundedness*, *conflicts*, *liveness*, *reversibility*, *persistency*, *deadlock-freeness, P-* and *T-invariants*) capture precedence relations and structural interactions between system components. Behavioral properties depend on, and are coupled with, the PN initial marking m_0. Structural properties are determined using the PN topological structure following matrix-based

Table 1. Basic PN module node (*P* and *T*) *explanation*

Node	*Model*	*Meaning*
p_0	All four modules	Parts available in initial buffer
p_1	All four modules	Machine available to process part
p_2	All four modules	Machine breakdown
p_3	All four modules	Parts in final buffer
p_4	All four modules	Machine finished process of a part
p_5	All four modules	Machine out of order
p_6	Assembly	Second type parts available in corresponding initial buffer (raw materials)
	Disassembly	Second type parts in the corresponding final buffer (processed parts)
	Parallel machines	Second Machine (M_2) available to process part
p_7	Parallel machines	Machine M_2 breakdown
p_8	Parallel machines	Machine finished process of a part
p_9	Parallel machines	Machine M_2 out of order
t_1	Common	Machine processing (producing) part
t_2	Common	Empty machine's signal return
t_3	Common	Machine breaks down
t_4	Common	Machine has been repaired and is available to produce again
t_5	Parallel machines	Machine M_2 is processing (producing) part
t_6	Parallel machines	Empty machine's signal return for M_2
t_7	Parallel machines	Machine M_2 breaks down
t_8	Parallel machines	Machine M_2 has been repaired and is available to produce again

analysis methods (Peterson, 1981; Murata, 1989; David and Aua, 1992; Reisig, 1992; Zhou and DiCesare, 1993; Desrochers and Al-Jaar, 1995; Proth and Xiaolan, 1996).

The incidence matrix A for a PN consisting of n_p places and n_t transitions is defined as $A=[a_{ij}]$, where $a_{ij}=a_{ij}^{+}-a_{ij}^{-}$; $a_{ij}^{+}=w\,(i, j)$ is the arc weight from transition i to its output place j and $a_{ij}^{-}=w(j,i)$ is the arc weight to transition i from its input place j. a_{ij}^{+}, a_{ij}^{-} and a_{ij} represent the tokens added, removed and totally changed in a place j by the firing of transition i, respectively. The incidence matrix cannot represent self-loops (since the total difference of tokens in a self loop is equal to 0).

The incidence matrix is used for the invariants calculation. Given a PN, there exists *place* or *P-invariants* and *transition* or *T-invariants*. *P-invariants* are the non-zero non-negative integer solutions X of the matrix equation $X^TA=0$ that also satisfy $X^Tm=X^Tm_0$ where X is an n_p-element vector, m_0 the initial marking of the net and m a marking that belongs to the reachability set of m_0, $R(m_0)$. *T-invariants* are the non-zero non-negative integer solutions Y of the matrix equation $AY = 0$ where Y is an n_t-element vector. There are (n_p-r) basic *P-invariants* and (n_t-r) *T-invariants*, where $r=rank(A)$. *P-invariants* express a notion of token conservation in sets of places for all reachable markings without enumeration of the reachability set $R(m_0)$. *T-invariants* describe a transition firing sequence S, such that $m_j \rightarrow\ m_j$. In consequence, any cyclic repetitive sequence of marking changes of a system represented by a PN is a *T-invariant*.

Since any linear combination of two or more *P-(T-)* invariants is also a *P-(T-) invariant*, only the sets of basic (or minimal) invariants are of interest and are calculated (Torn). The Martinez – Silva algorithm (Martinez and Silva, 1982) may be utilized to determine net invariants by removing consecutive net elements.

Dealing with time in a PN is accomplished by assigning time delays to places or transitions. Time can be associated with both nodes, but TPN models analysis is simpler when time is attached to one kind of nodes (Mellado, 2002). The case described is that of PNs where time is attached to transitions, called *t*-timed *PNs* nets.

A *t*-timed PN arises from the corresponding Ordinary PN by associating each transition t_i a firing delay that may be constant or follow a given distribution. A TPN is defined by the tuple $TPN=\{P, T, I, O, m_0, D,\}$ such that the first five elements are as described above for ordinary PNs while D represents time delay and is a function from the set of non-negative real numbers $\{0,R^+\}$. $D(t_i)$ is a vector whose number of elements is the same with the number of net's transitions, where d_i = delay associated with transition i. A timed transition's firing consists of two events namely, "start firing" and "end firing". In between these two events the firing is in progress. Tokens are removed from input places at "start firing" and are deposited to output places at "end of firing" (Venkatesh and Ilyas, 1995; Zuberek, 2001). The transitions delays may be deterministic or described by a distribution. In TPNs some transitions may have zero occurrence times and are called "immediate".

All TPN properties and analysis methods used are the same with the ones used for the corresponding OPNs (David and Alla, 1992; Boel and Stremersch, 1999). Though, if the analysis of a TPN is tried by using the corresponding OPN model, wrong conclusions for some issues may be obtained.

3.2. *Generic Petri Net modules and invariant calculation*

The PN modules of Fig. 1, called *generic PN modules* from now on, are shown in Figs. 3–6. These modules arise from the manufacturing system control modules defined in Tsourveloudis *et al.* (2000) and Ioannidis *et al.* (2002). PN models describe all the main events taking place in fundamental systems and are suitable for production systems simulation, analysis and performance evaluation. Modules describe common events but partially differ in operational and structural features. Timed transitions are presented as white rectangles, while immediate transitions as black. All transition input and output arc weights are equal to one. Table 1 explains places and transitions meanings. Places p_0–p_5 and transitions t_1–t_4 have the same meaning in all four generic PNs. Transitions correspond to system activities resulting in state changes, while places correspond to resource (machine, parts) availability or state (machine up, down, idle).

The generic PN modules have been derived based on the realistic assumptions that: (i) buffers are finite, (ii) machines operate at a given speed that changes periodically according to events taking place in the system, (iii) setup and transportation times of parts through the system are negligible compared to production times, (iv) machine breakdowns happen infinitely often, but only after the completion of a production cycle.

The generic transfer chain module is considered first. In this, places p_0 and p_3 represent the raw materials and final products buffers respectively, while t_1 describes process performance when machine is available. To avoid multiple parts reaching a machine concurrently, a signal of machine being empty and ready to produce is represented through a token produced from the firing of t_1 and led to p_4. Empty machine's signal returns to p_1, where it is initially found, meaning that next parts process may begin, following one of two possible routes. One is after firing of t_2, meaning that the next part process can begin without other events mediation. The second is after a machine breakdown represented by p_2, t_3 fires causing machine to go out of order (p_5), which is repaired after firing of t_4. In all generic and generalized PN modules, conflict between t_2 and t_3 that have common input place p_4, exists – machine finished a part process (in modules that have

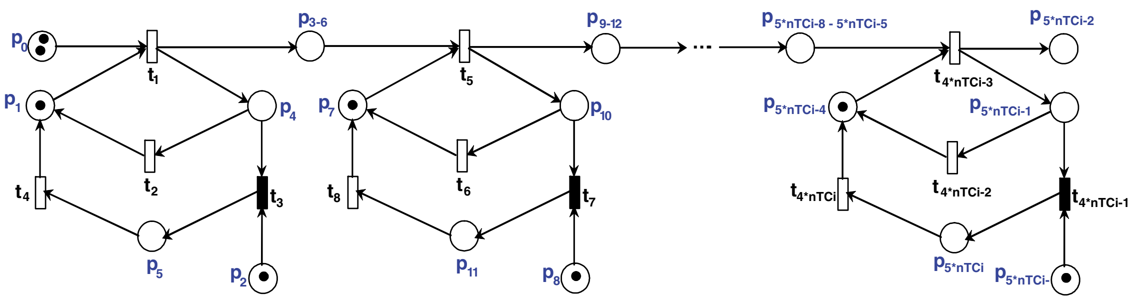

Fig. 7. Generalized transfer chain PN module.

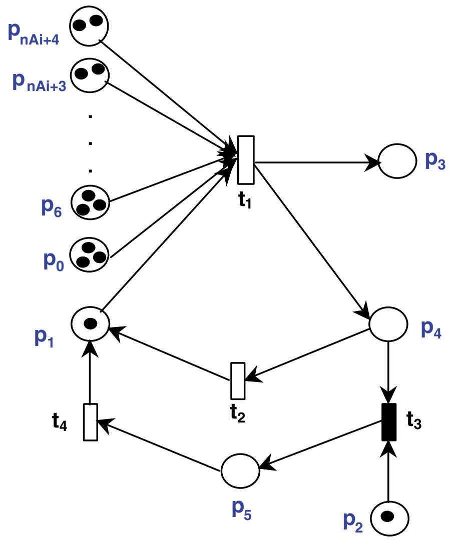

Fig. 8. Generalized assembly PN module.

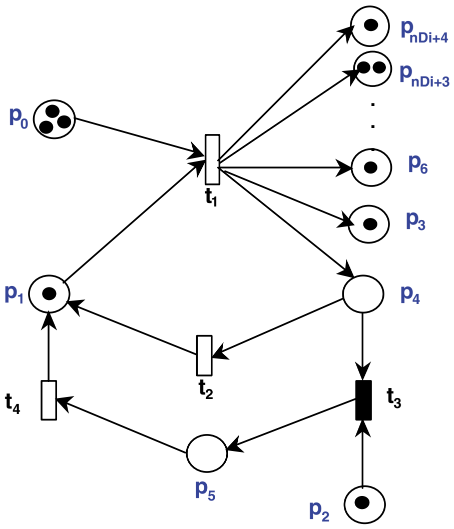

Fig. 9. Generalized disassembly PN module.

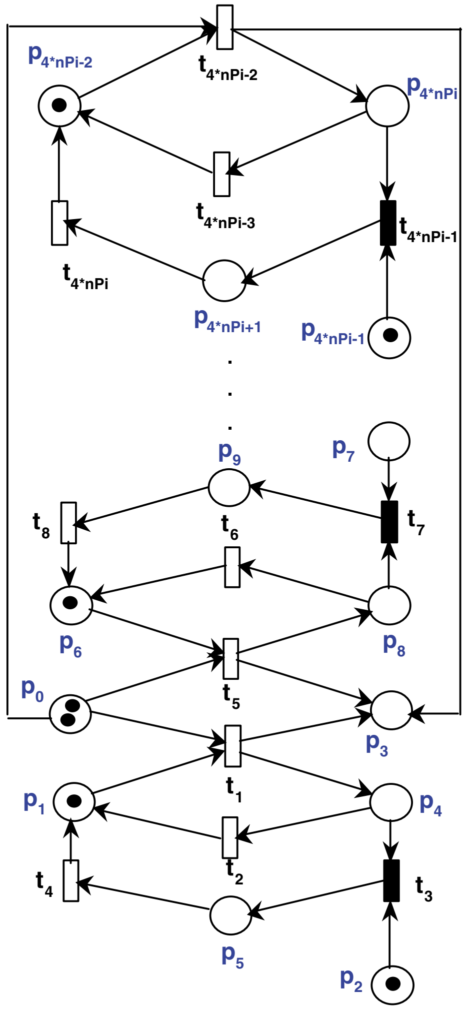

Fig. 10. Generalized parallel machines PN module.

multiple machines, there exist proportional conflicts in all sets of them). To solve this conflict efficiently, t_2 has been considered as a timed transition (with minimum delay) while t_3 as an immediate one. t_3 has also a second input place (p_2) (tokens are available there only in the case of a machine breakdown according to a distribution or stochastically), which means that it cannot become enabled and consequently fire if there exist no tokens in this place. So, when machine operates, timed transition t_2 fires, since immediate t_3 has no token in one input place, else t_3 fires. It must be noted that in the models presented both transitions have the same priority (equal to one), since in another case the execution of the net would be different. Generic assembly module's main difference is that two initial buffers (p_0, p_6) provide different part types to the machine for the process to begin, while generic disassembly module has two final product buffers (p_3, p_6). In generic parallel machine module two transfer chain PN modules are composed so that they obtain parts from the same input buffer (p_0) and lead their products to the same final buffer (p_6). These facts are generalized for the cases of generalized PN modules. Definition of buffers maximum capaci-

Table 2. Node complexity of generalized PN modules

Model	*Node type*	*Generic model*	*Generalized model*
Transfer chain	*P*	6	$5^{*}\ n_{TCi}+1$
	T	4	$4^{*}\ n_{TCi}$
Assembly	*P*	7	$n_{Ai}+5$
	T	4	4
Disassembly	*P*	7	$n_{Di}+5$
	T	4	4
Parallel machines	*P*	10	$2+4^{*}\ n_{Pi}$
	T	8	$4^{*}\ n_{Pi}$

ties in PN modules ensures the representation of phenomena as starved and blocked machines. Tokens shown in all four generic PN modules are for demonstration purposes only.

Observing the four generic PN modules as shown in Fig. 3–6 with any finite initial marking m_0, one may conclude that: (i) As long as there is part availability in the input buffer(s), all four generic PNs after the completion of one production cycle return to the state of starting a new cycle; (ii) the parts number in the initial buffer(s) defines the exact number of production cycles; (iii) all modules are k- bounded; (iv) modules are non-conservative (at least t_3 consumes two tokens and produces one, in assembly and disassembly transition t_1 is also non-conservative, in parallel machines module t_7 is non-conservative); (v) modules are non-persistent (firing of t_3 may disable t_2); (vi) modules are not repetitive and not consistent. Further the following hold:

1. For the transfer chain, the upper limit of tokens found in a place, UL_{TC}, is defined as $min\{max\ \{C_0,\ C_3\},(m_0(p_0)+m_0(p_3))\}$, where C_i is the maximum capacity of p_i. That is, the maximum number of tokens in a place is the minimum of two quantities, maximum capacity of the two buffers and the sum of the initial tokens in these two places.
2. For the assembly module, the upper limit of tokens found in a place, UL_A, is slightly different since there are at least two input buffers, defined as $min\ \{max\ \{C_0,\ C_3,\ C_6\},(m_0(p_3)+max\ \{m_0\ (p_0),\ m_0(p_6)\})\}$.
3. For the disassembly module, the upper limit of tokens found in a place, UL_D, is calculated considering two output buffers as $min\{max\ \{C_0,\ C_3,\ C_6\},(m_0(p_0)+max\{m_0\ (p_3),\ m_0(p_6)\})\}$.
4. For the two parallel machines module, the corresponding equation, UL_P, is the same with the one of transfer chain, $min\ \{max\ \{C_0,\ C_3\},(m_0(p_0)+m_0(p_3))\}$, since the same buffers are used for both machines.
5. Except the places that represent buffers, all other places are safe.

The Martinez–Silva algorithm (Martinez and Silva, 1982) is used to calculate the minimal *P*- and *T-invariants* (all other invariants are linear combinations of minimal invariants). The transfer chain module has two *P-invariants* and no *T-invariant*. The *P-invariants* are:

$$\{100100\}\ \text{resulting in}\ m(p_0)+m(p_3)=k_1$$

$$\{010011\}\ \text{resulting in}\ m(p_1)+m(p_4)+m(p_5)=1$$

$$k_1=m_0(p_0)+m_0(p_3)$$

The first *P-invariant* guarantees that the sum of parts in the initial and in the final buffer is constant and equal to the initial sum of parts in these two buffers k_1, while the second shows three mutually exclusive machine states (machine ready to process part, empty or machine breakdown). Those two *P-invariants* are common for all four generic PN modules. However, the other three modules have one more *P-invariant*. The third *P-invariant* of the assembly module refers to the sum of tokens of the second initial buffer and final buffer that is equal to the initial sum of parts in these two buffers:

$$m(p_6)+m(p_3)=k_2,\ k_2=m_0(p_6)+m_0(p_3)$$

The third *P-invariant* of the disassembly module refers to the sum of tokens of the initial buffer and the second final buffer:

$$m(p_0)+m(p_6)=k_3,\ k_3=m_0(p_0)+m_0(p_6)$$

The third *P-invariant* of the two parallel machines module refers to the mutually exclusive states of the second machine M_2, same with the ones of the first machine:

$$m(p_6)+m(p_8)+m(p_9)=1$$

3.3. *Generalized Petri Net modules and invariant calculation*

The four general PN models that correspond to the four modules of Fig. 2, called *generalized PN modules* from now on, are shown in Figs. 7–10. The same assumptions made for the generic PN modules are considered here, too, and the same observations may be made. Table 2 shows PN module complexity, in terms of places and transitions, for the simplest and most general case.

The generalized transfer chain has $(n_{TCi}+1)$ *P-invariants* and no *T-invariant*. One *P-invariant* refers to the preservation of tokens (parts) within the system (with k_1 the sum of tokens in these places, initially) while the rest refer to the mutually exclusive states of the machines of the generalized transfer chain (one for each machine). The *P-invariants* are:

$$m(p_0) + m(p_{3-6}) + m(p_{9-12}) + \cdots + m(p_{5*nTCi-8-5*nTCi-5}) + m(p_{5*nTCi-2}) = k_1$$

$$m(p_1) + m(p_4) + m(p_5) = 1$$

$$m(p_7) + m(p_{10}) + m(p_{11}) = 1$$

$$\ldots$$

$$m(p_{5*nTCi-4}) + m(p_{5*nTCi-1}) + m(_{5*nTCi}) = 1$$

The generalized assembly module has $(n_{Ai}+1)$ *P-invariants* and no *T-invariants*. One *P-invariant* refers to the mutually exclusive states of the machine while the rest refer to the preservation of tokens within the module, one *P-invariant* for each input buffer. Knowing that k_j, $j=1,\ldots,n_{Ai}$ represents the sum of tokens in the corresponding places given the initial marking m_0, the *P-invariants* are:

$$m(p_1) + m(p_4) + m(p_5) = 1 \text{(mutually exclusive states of the machines)}$$

$$m(p_0) + m(p_3) = k_1$$

$$m(p_6) + m(p_3) = k_2$$

$$\ldots$$

$$m(p_{nAi+3}) + m(p_3) = k_{nAi-1}$$

$$m(p_{nAi+4}) + m(p_3) = k_{nAi}$$

The generalized disassembly module has $(n_{Di}+1)$ *P-invariants* and no *T-invariants*. One *P-invariant* refers to the mutually exclusive states of the machine while the rest refer to the preservation of tokens within the module (one *P-invariant* for each output buffer). These are (again, k_j, $j=1,\ldots,n_{Di}$ represents the sum of tokens in the corresponding places given the initial marking m_0):

$$m(p_1) + m(p_4) + m(p_5) = 1 \text{(mutually exclusive states of the machines)}$$

$$m(p_0) + m(p_3) = k_1$$

$$m(p_0) + m(p_6) = k_2$$

$$\ldots$$

$$m(p_0) + m(p_{nDi+3}) = k_{nDi-1}$$

$$m(p_0) + m(p_{nDi+4}) = k_{nDi}$$

Finally, the generalized parallel machine module has $(n_{Pi}+1)$ *P-invariants* and no *T-invariants*. One *P-invariant* refers to the preservation of tokens within the module while the rest refer to the mutually exclusive states of the machine, one *P-invariant* for each of the parallel machines):

$$m(p_0) + m(p_3) = k_1 \text{(with } k_1 \text{ the sum of tokens in these places initially)}$$

$$m(p_1) + m(p_4) + m(p_5) = 1 \text{(mutually exclusive states of machine } M_1\text{)}$$

$$m(p_6) + m(p_8) + m(p_9) = 1 \text{(mutually exclusive states of machine } M_2\text{)}$$

$$\ldots$$

$$m(p_{4*nPi}) + m(p_{4*nPi+1}) + m(p_{4*nPi-2}) = 1 \text{ (mutuallyexclusive states of machine } M_{nPi}\text{)}$$

3.4. *Theoretical calculation of the generalized PN module invariants*

Considering the overall system PN model, *P-invariants* can be considered of two types, one referring to the mutually exclusive states of the machines, the other referring to the preservation of parts number. From the first type there are d_1 *P-invariants*, where d_1 is equal to the number of production system machines, calculated with respect to the number of modules of each type used and the components of each module. So d_1 is given by

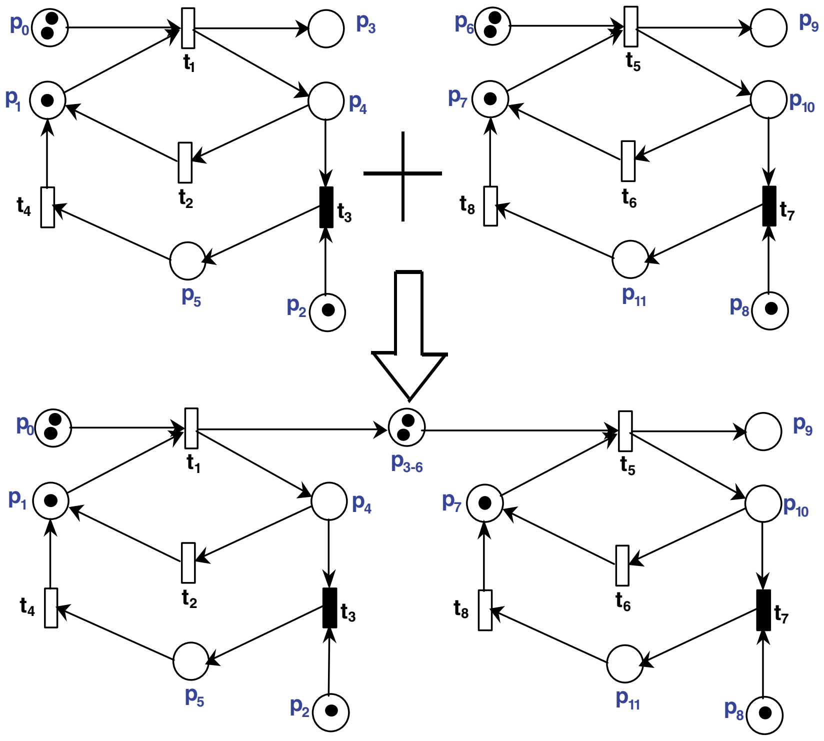

Fig. 11. Synthesis of two generic transfer chain modules..

$$d_1 = \sum_{i=1}^{n_1} n_{TC_i} + n_2 + n_3 + \sum_{j=1}^{n_4} n_{P_j}$$

From the second type (reservation of parts number) there are d_2 *P-invariants*. For calculation of d_2 two distinct cases are examined according to nets constitution that produce two different equations regarding their structure and complexity. In the first case, the manufacturing system does not perform assembly and disassembly operations in parallel (this is mathematically translated in $n_2 * n_3 = 0$), independently of how complicated its topology and structure is (this case refers to production systems consisting of sets of transfer chain, parallel machine and assembly modules, or transfer chain, parallel machines and disassembly modules, or simpler cases where assembly and disassembly do not exist in the same net). In this case d_2 is:

$$d_2 = n_E + \sum_{j=1}^{n_3} (n_{Dj} - 1) = n_E{}^* n_F$$

This equation shows that d_2 *P-invariants* are equal to the number of the external (non-fused) input places of the net increased by the total number of extra buffers produced by disassemblies (each disassembly module "produces" two or more buffers from one initial, which means that a least one extra buffer is generated). In this case number of d_2 P-invariants is also equal to the product of the input and output buffers of the net (both equations give the same result and are independent each-other.)

When both assembly and disassembly modules exist in a manufacturing system, the equation presented above cannot describe a number of internal events taking place in the system. In particular, when there are assemblies and disassemblies "sequentially" in a net (assembly after disassembly or vice-versa), some of the P-invariants of the net are not calculated when using the equation above. In this case the equation that calculates d_2 is much more complicated and comes out by regarding for every initial part the sequence of operations received from its input (external buffer) until its exit (final buffer(s)) from the system. For such a system, d_2 is:

$$\begin{aligned} d_2 = \sum_{i=1}^{n_E} d_{2,i} = \sum_{i=1}^{n_E} \Bigg[& n_{D_{i,1}} - \sum_{i_1=1}^{c} n_{A_{1,i_1}} \\ & + n_{A_{1,1}}{}^* \left[n_{D_{i,2,1}} - \sum_{i_{21}=1}^{c_{21}} n_{A_{2,i_{21}}} + n_{A_{2,1}}{}^* \left[n_{D_i,3,1} - \dots \right] \right] \\ & + \dots + n_{A_{1,c_1}}{}^* \Bigg[n_{D_{i,2,c_1}} - \sum_{i_{2c_1}=1}^{c_{2c_1}} n_{A_{2,i_{2c_1}}} \\ & + n_{A_{2,c_1}}{}^* \left[n_{D_{i,3,c_1}} - \dots \right] \Bigg] \Bigg] \end{aligned}$$

For every initial part n_E, $d_{2,i}$ number of it's *P*-invariants is calculated. In this, number of the first received disassembly products, is considered ($n_{Di,1}$). This number is decreased by the number of these products participating in second level assemblies, represented by the sum (second level starts after the initial disassembly and ends before second disassembly). In this, c_i refers to assemblies in which the initial disassembly products participate. $d_{2,i}$ is increased for every second level assembly by adding the product of input to assembly parts ($n_{A1,1}$) and the piece of the equation that already has been described, adapted to third level features (number of second level disassembly products − sum of parts participating in third level assemblies + product of number of parts participating in third level assemblies * the piece of the equation that has been described, adapted to fourth level characteristics). This iterative process is repeated for as many levels as the net has (each level is defined by two sequential disassemblies, the first defines level's starting point which ends just before the second). If an initial part does not take part in any disassembly, the respective factor in the equation is set equal to one.

4. Petri Net module synthesis

The synthesis procedure of two simplest transfer chains is shown in Fig. 11. Generalizations follow.

Observing Fig. 11, it is obvious that places p_3 and p_6 are fused in one place $p_{3\text{–}6}$. The total number of places is reduced by one, while transitions are equal to the total of each module transitions. The combined PN input places are reduced by one ($p_{3\text{–}6}$ is an internal place, not input buffer). The maximum capacity of $p_{3\text{–}6}$ may be defined as $C_{3\text{–}6} = \min\{C_3, C_6\}$ or $C_{3\text{–}6} = \max\{C_3, C_6\}$ or with any number in between (based on system constraints). Obviously, $m_0(p_{3\text{–}6}) = m_0(p_3) + m_0(p_6)$.

The combined PN properties may be detected accordingly, by simple observation. There exist three *P-invariants* (two are identical with the individual module *P-invariants*). Two refer to the mutually exclusive states of the combined PN; the third refers to the preservation of the total number of parts in the PN, where k_1 is the initial sum of parts (tokens) in the three places:

$$m(p_1) + m(p_4) + m(p_5) = 1$$

$$m(p_7) + m(p_{10}) + m(p_{11}) = 1$$

$$m(p_0) + m(p_{3\text{–}6}) + m(p_9) = k_1$$

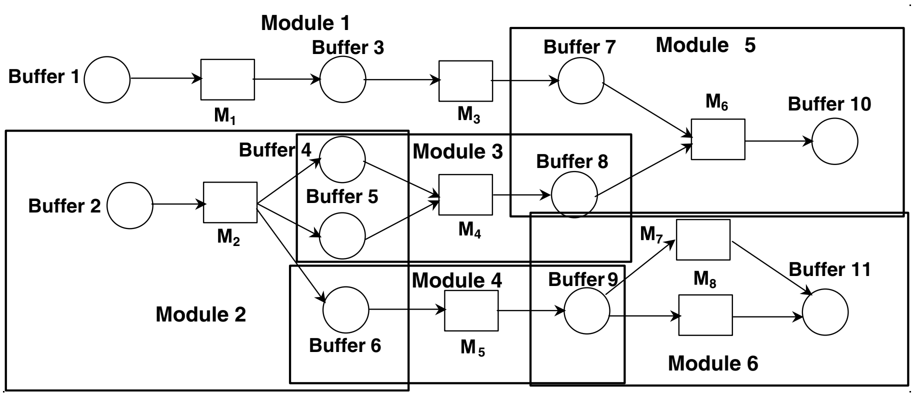

Fig. 12. A generalized Production System and its module decomposition.

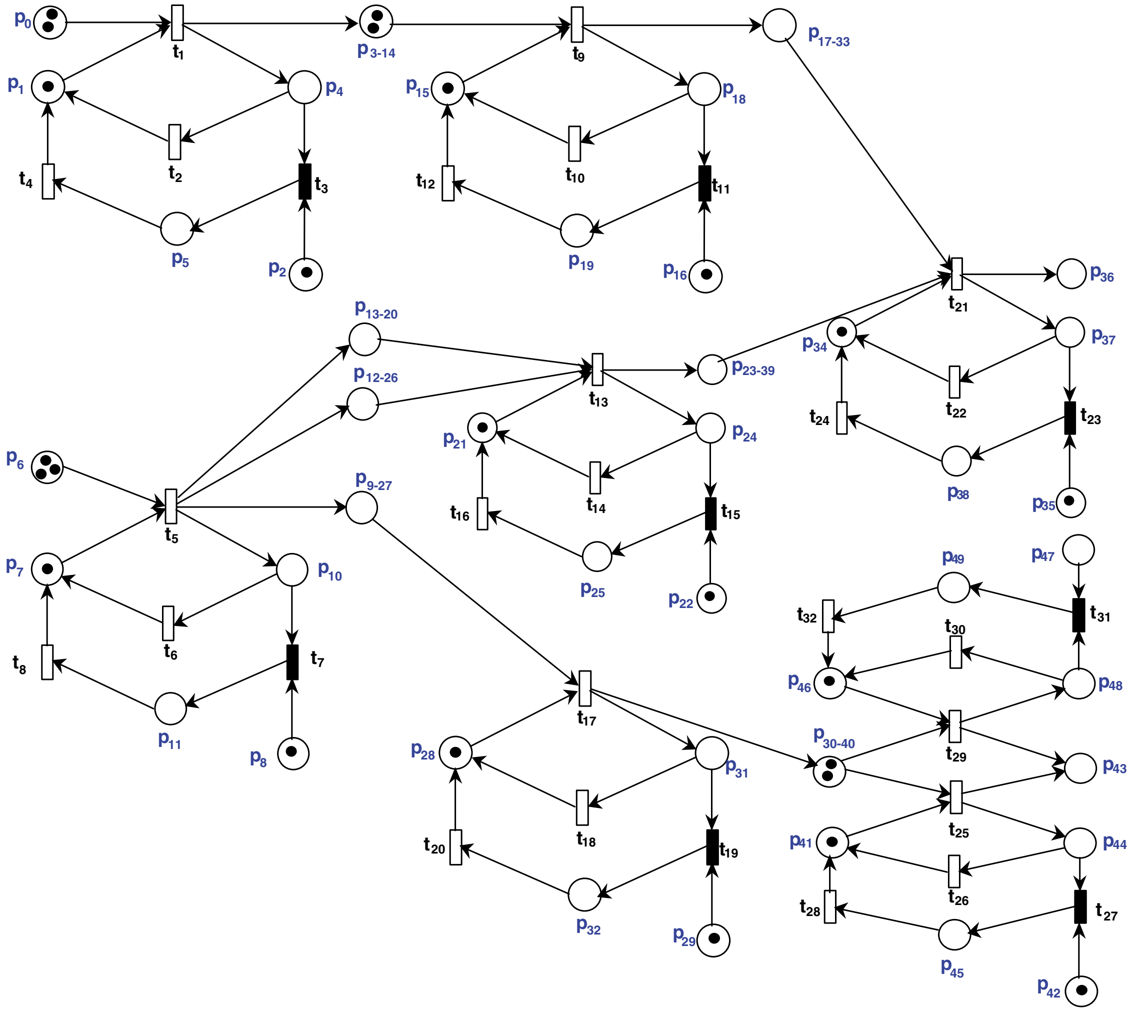

Fig. 13. The Petri net model of the production system of Fig. 12.

Synthesis of other generic PN modules is obtained in a similar way.

4.1. *Generalizations*

It is possible to calculate the number of nodes of a random topology production system PN model from the corresponding PN modules that compose it and from the number of external parts entering the system. The latter number is necessary to compute the number of fused places in the individual modules connection points.

Consider first that the overall production system PN model is derived in terms of the generic PN modules of Figs. 3–6. That is, consider n_1 modules of transfer chain, n_2 modules of two-parts assembly, n_3 modules of two-piece disassembly, n_4 modules of two-parallel machines. In addition assume that there are n_E (external, non-fused) input places and n_F final buffers. The number of the combined PN model transitions is:

$$4^*(n_1 + n_2 + n_3) + 8^*n_4$$

The total number of generic PN module places when considered separately (no fused places) is:

$$6^*n_1 + 7^*(n_2 + n_3) + 10^*n_4$$

Fusion of places at connection points reduces the number of places calculated above. It must be noted, that in places fusion only places representing net's buffers participate. So, first of all the number

Table 3. Case study production systems timed transition parameters

Transition	*Distribution followed*	*Delay lower bound*	*Delay upper bound*
t_1	Deterministic	3	–
t_2	Deterministic	1	–
t_3	–	–	–
t_4	Uniform	1	3
t_5	Deterministic	5	–
t_6	Deterministic	1	–
t_7	–	–	–
t_8	Uniform	2	5
t_9	Deterministic	4	–
t_{10}	Deterministic	1	–
t_{11}	–	–	–
t_{12}	Uniform	1	5
t_{13}	Deterministic	2	–
t_{14}	Deterministic	1	–
t_{15}	–	–	–
t_{16}	Uniform	3	5
t_{17}	Deterministic	3	–
t_{18}	Deterministic	1	–
t_{19}	–	–	–
t_{20}	Uniform	1	7
t_{21}	Deterministic	4	–
t_{22}	Deterministic	1	–
t_{23}	–	–	–
t_{24}	uniform	1	3
t_{25}	deterministic	2	–
t_{26}	deterministic	1	–
t_{27}	–	–	–
t_{28}	uniform	2	3
t_{29}	deterministic	4	–
t_{30}	deterministic	1	–
t_{31}	–	–	–
t_{32}	Uniform	1	5

of net's buffers for a production system consisting of n_1 transfer chain modules, n_2 assemblies, n_3 disassemblies, n_4 parallel machine modules and that has n_E external input buffers and n_F final (product) buffers is calculated. That is $2{*}n_1+3{*}n_2+3{*}n_3+2{*}n_4$, since transfer chain and parallel machine modules have one input and one output place, assembly has two input and one output place and disassembly has one input and two output places. The total net's model has n_E input places and n_F output places, since all the other places are any more internal. So these n_E+n_F places are not fused and are subtracted. From the remaining, from each two places one fused is generated, so the total number of fused places is:

$$1/2 * (2 * n_1 + 3 * n_2 + 3 * n_3 + 2 * n_4 - n_\mathrm{E} - n_\mathrm{F})$$

n_F can be calculated with respect to n_E as n_F is equal to the external buffers number reduced by one for each assembly and increased by one for each disassembly. So:

$$n_\mathrm{F} = n_\mathrm{E} - n_2 + n_3$$

Thus, the number of fused places is calculated as:

$$1/2^*(2^*n_1 + 3^*n_2 + 3^*n_3 + 2^*n_4 - n_\mathrm{E} - n + n_2 - n_3)$$
$$= n_1 + n_3 + n_4 + 2^*n_2 - n_\mathrm{E}$$

The total number of the combined net places is:

$$5^*(n_1 + n_2) + 6^*n_3 + 9^*n_4 + n_\mathrm{E}$$

Considering individual generalized PN modules as shown in Table 2, the total number of transitions is:

$$4^*(n_2 + n_3) + \sum_{i=1}^{n_1} 4^*n_{TCi} + \sum_{j=1}^{n_4} 4^*n_{Pj}$$

The total number of places is calculated as:

$$\sum_{i_1=1}^{n_1}(5^*n_{TCi_1} + 1) + \sum_{i_2=1}^{n_2}(n_{Ai_2} + 5) + \sum_{i_3=1}^{n_3}(n_{Di_3} + 5)$$
$$+ \sum_{i_4=1}^{n_4}(4^*n_{Pi_4} + 2) - (n_1 + n_3 + n_4 + \sum_{i_2=1}^{n_2}(n_{Ai_2} - n_E)$$
$$= \sum_{i_1=1}^{n_1} 5^*n_{TCi_1} + \sum_{i_3=1}^{n_3} n_{Di_3} + \sum_{i_4=1}^{n_4} 4^*n_{Pi_4} + 5^*n_2$$
$$+ 4^*n_3 + n_4 + n_E$$

5. Case studies and simulation results

The production system of Fig. 12 is used as a case study. It is composed of six modules, from which two are generalized transfer chains, one parallel machines module, two assemblies and one disassembly. The transfer chain module one consists of two sequential machines and the disassembly module two creates three parts from one. The production system as shown in Fig. 12 consists of 11 buffers and 8 machines.

The overall PN module is shown in Fig. 13. The PN consists of 43 places and 32 transitions. Eight transitions are immediate corresponding to potential machine breakdowns. There are two

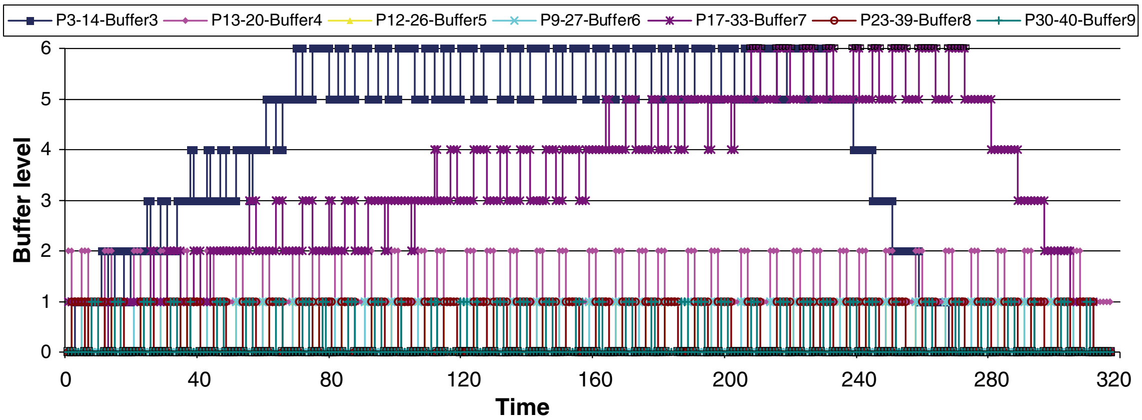

Fig. 14. Internal buffer levels during initial simulation.

external input places p_0 and p_6. There are two places at the system where parts reaching them are considered as finished parts of different types (products). These places are p_{36} and p_{43} and correspond to buffers 10 and 11. Also there are seven fused places that represent internal buffers of the system at the connection points of the individual modules.

There are 8 *P-invariants*, each referring to the mutually exclusive states of a machine: $m(p_1)+m(p_4)+m(p_5)=1$ for M_1, $m(p_7)+m(p_{10})+m(p_{11})=1$ for M_2, $m(p_{15})+m(p_{18})+m(p_{19})=1$ for M_3, $m(p_{21})+m(p_{24})+m(p_{25})=1$ for M_4, $m(p_{28})+m(p_{31})+m(p_{32})=1$ for M_5, $m(p_{34})+m(p_{37})+m(p_{38})=1$ for M_6, $m(p_{41})+m(p_{44})+m(p_{45})=1$ for M_8 and $m(p_{46})+m(p_{48})+m(p_{49})=1$ for M_7. In addition, there are 4 *P-invariants* referring to the preservation of the parts number within the system: $m(p_0)+m(p_{3\text{-}14})+m(p_{17\text{-}33})+m(p_{36})=k_1$, $m(p_6)+m(p_{13\text{-}20})+m(p_{23\text{-}39})+m(p_{36})=k_2$, $m(p_6)+m(p_{12\text{-}26})+m(p_{23\text{-}39})+m(p_{36})=k_3$ and $m(p_6)+m(p_{9\text{-}27})+m(p_{30\text{-}40})+m(p_{43})=k_4$ where k_1, k_2, k_3 and k_4 is the sum of tokens in the corresponding sets of places in the initial marking m_0. Furthermore, tokens in places p_2, p_8, p_{16}, p_{22}, p_{29}, p_{35}, p_{42} and p_{47} indicate the appearance of machine breakdowns (that may be random or follow a given distribution).

With initial marking $m_0=\{40, 1, 0, 0, 0, 0, 40, 1, 0, 0, 0, 0, 1, 2, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0\}$ and timed transition parameters as shown in Table 3, the simulation of the net of Fig. 13 is completed after 836 steps and has a total duration of 317 time

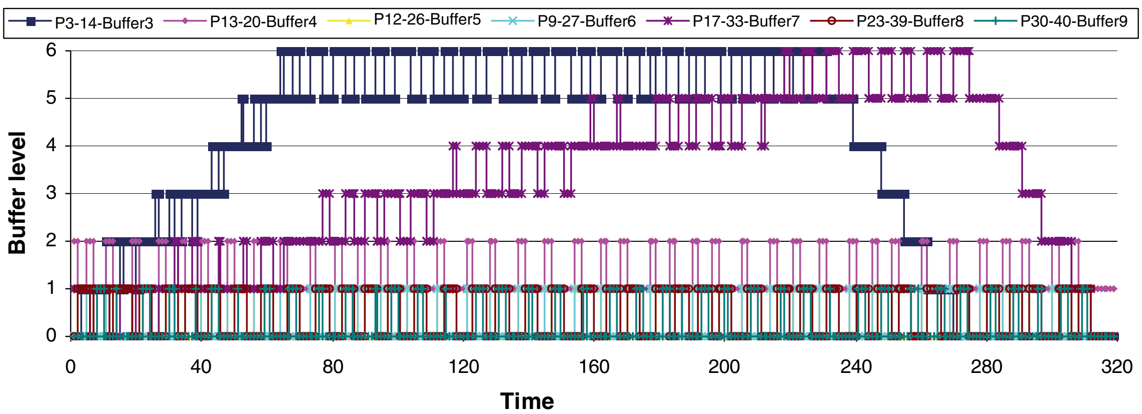

Fig. 15. New internal buffer levels after the reduction of the capacities of 5 buffers.

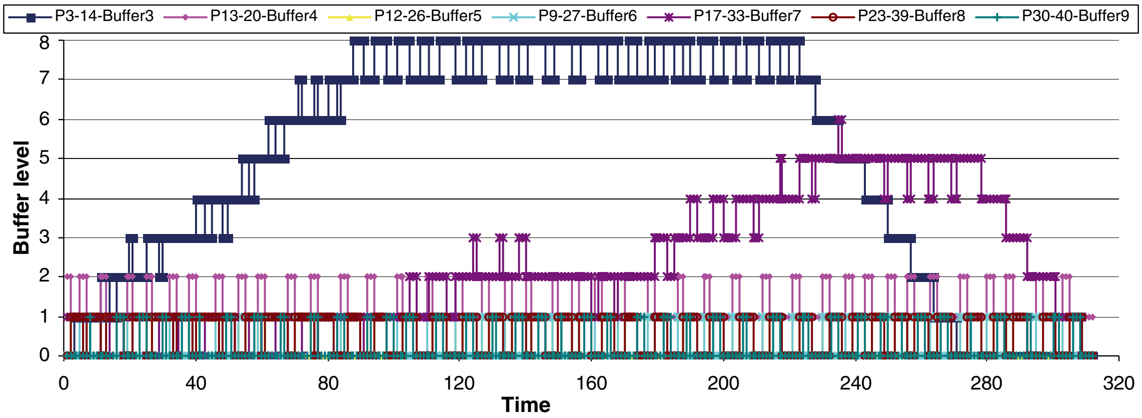

Fig. 16. Internal buffer levels after the increase of the capacities of buffers 3 and 7.

units. After the completion of the simulation, 41 parts are found in the final buffer p_{36} (*type 1* parts) and 40 in the final buffer p_{43} (*type 2* parts). The final marking of the net with initial marking m_0 is: $m_f = \{0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 41, 0, 0, 1, 1, 40, 0, 0, 1, 1, 0, 0\}$. The final marking m_f is common for all the simulations performed, as the parameters that change have to do with the net characteristics that have impact in the product cycle time (machine processing speeds, buffer capacities, mean breakdown durations etc.) and not with the number of parts processed. The mean production time for *type* parts is 7.732 time units and for *type* 2 parts 7.925 time units. Fig. 14 shows the internal buffer levels during the simulation, in order to become clear the mean usage of each buffer so that the real capacities of the buffers can be optimized.

In initial simulation, in parallel machines module machine M_7 produces 16 parts while M_8 the remaining 24 (not in all cases the two parallel machines process the same number of parts, as the number of parts led to each depends on their state when new parts reach the common input buffer, on the relative speeds of the machines before and after and on other parameters). As expected the % operating time of the machines has a strong relation with the position of the machine in the net's structure, i.e. the first and last machines have lower percentages since they have long periods of idleness in the end and in the beginning of net's operation respectively. The % operating time refers to the proportion of the total time (317 time units) that each machine is processing parts. The % operational time refers to the proportion of the total time that a machine is available to process parts, but in a percentage of this machine remains idle since it has no parts available. The difference between operational and operating time calculates the proportion of time that machine remains idle.

From Fig. 14, it is obvious that the only two internal buffers that their levels reach their capacities during the simulation are buffers 3 (represented by place p_{3-14}) and 7 (p_{17-33}). Buffer 3 works at capacity for longer time period that buffer 7, while the rest five buffers do not have more than two parts during the simulation.

An interesting point is the repetition of the simulation with the features presented above but without considering machine breakdowns (by considering machines operational for the whole simulation time). In this case the simulation is completed in 501 steps and has a total duration of 246 time units. This means that the mean production time for *type 1* products is 6 time units and for *type 2* parts the corresponding time is 6.15 time units. In this case, in parallel machines module machine M_7 produces 18 products while M_8 the remaining 22 (this arises from the simulation). Also in this case the calculation of machines operational time has no meaning since all machines are operational in 100% of their time and remain idle in the proportion of time that they do not produce. By comparing these numbers with the production times of the initial simulation it is obvious that the impact of the machine breakdowns is very important and if they are omitted the results obtained are far from reality.

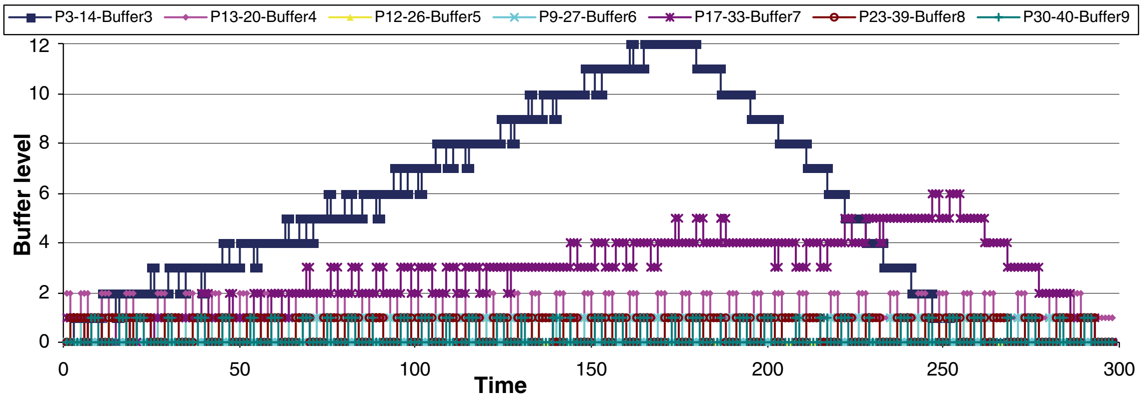

Fig. 17. Buffer levels during infinite buffers simulation.

The increase of the production times due to machine breakdowns is of order of 28%. The production times calculated by this change can never in fact be succeeded, but give a bound for the calculated parameters.

A first optimisation step concerns the reduction of the internal buffers capacities that do not reach their maximum capacities during the simulation. These changes should not affect notably the duration of the new simulation, as the new capacities are bigger than the maximum number of parts founded there during initial simulation. The new capacities of buffers p_{9-27}, p_{12-26}, p_{13-20}, p_{23-39} and p_{30-40} are reduced to half and are equal to 3 parts. The internal buffer levels after the changes are presented in Fig. 15.

It is obvious that Figs. 14 and 15 are almost identical, as the capacities that have been changed are not crucial since number of parts in these buffers did not reach this limit. In this simulation, in parallel machines module machine M_7 produces 14 parts while M_8 the remaining 26 parts. In this case the simulation is completed after 843 steps and has a total duration of 319 time units. The mean production time for *type 1* parts, is 7.78 time units and for *type 2* parts 7.975 time units, approximately equal to initial production times.

Next step concerns the increase of buffers 3 and 7 capacities that reach their previous maximum capacities for significant proportions of the operating time. Their new capacities are set to 8 and 7 since the first one is full for longer time intervals. After this change, the respective buffer levels are presented in Fig. 16.

From Fig. 16, it can be seen that only the level of buffer 3 reaches its capacity for a significant portion of the operational time of the system. This means that a blockage takes place in this part of

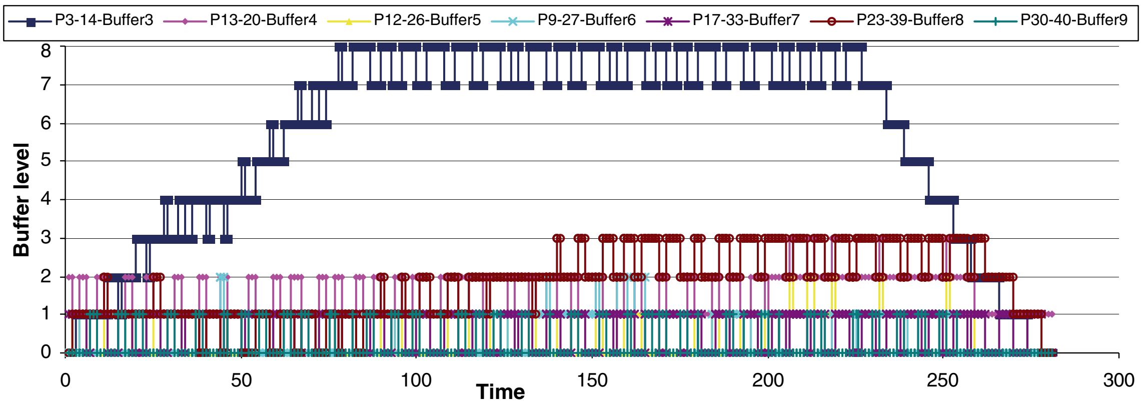

Fig. 18. Buffer levels after the reduction of the production time of M_2.

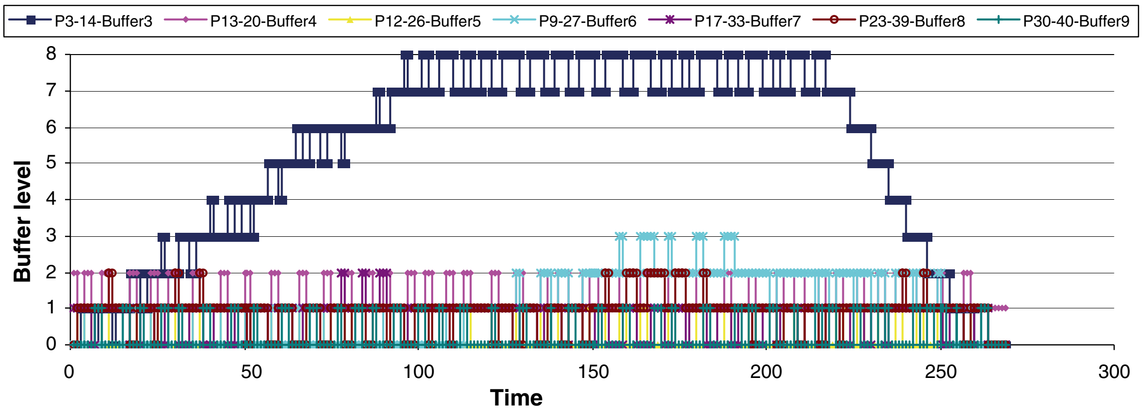

Fig. 19. Buffer levels after the reduction of breakdown appearances in machine M_5.

the system and a further improvement is necessary. The simulation is completed after 817 steps with total duration of 312 time units. There is a little improvement in the operation of the net as there is a reduction of 5 time units of the total operational time in comparison with the initial simulation. In this simulation, in parallel machines module machine M_7 produces 17 parts while M_8 the remaining 23. The mean production time for *type 1* parts is 7.61 time units and for *type 2* parts 7.8 time units. The improvement is about 2%.

If all internal buffers (p_{3-14}, p_{9-27}, p_{12-26}, p_{13-20}, p_{17-33}, p_{23-39} and p_{30-40}) are considered of infinite capacity (practically for this case 40 parts as this is the number of parts led to the system) and the simulation is repeated with the same parameters interesting results are obtained. This allows the calculation of the maximum number of parts

Table 4. Calculation of multiple quantitative characteristics used for performance evaluation and optimisation of the manufacturing system for the seven simulations described above.

Parameter	*Simulation*						
	Initial	*Second*	*Third*	*Fourth*	*Fifth*	*Sixth*	*Seventh*
BUFFER i CAPACITY							
Buffer 1 (p_0) capacity	50	50	50	50	50	50	50
Buffer 2 (p_6) capacity	50	50	50	50	50	50	50
Buffer 3 (p_{3-14}) capacity	6	6	6	8	40	8	8
Buffer 4 (p_{13-20}) capacity	6	6	3	3	40	3	3
Buffer 5 (p_{12-26}) capacity	6	6	3	3	40	3	3
Buffer 6 (p_{9-27}) capacity	6	6	3	3	40	3	3
Buffer 7 (p_{17-33}) capacity	6	6	6	7	40	7	7
Buffer 8 (p_{23-39}) capacity	6	6	3	3	40	3	3
Buffer 9 (p_{30-40}) capacity	6	6	3	3	40	3	3
Buffer 10 (p_{36}) capacity	50	50	50	50	50	50	50
Buffer 11 (p_{43}) capacity	50	50	50	50	50	50	50
Duration							
Number of steps	836	501	843	817	775	725	719
In time units	317	246	319	312	298	281	269
Number of produced products							
Type 1 products	41	41	41	41	41	41	41

Table 4. (Continued)

Parameter	*Simulation*						
	Initial	*Second*	*Third*	*Fourth*	*Fifth*	*Sixth*	*Seventh*
Type 2 products	40	40	40	40	40	40	40
Mean production time							
Type 1 products	7.732	6	7.78	7.61	7.26	6.85	6.56
Type 2 products	7.925	6.15	7.975	7.8	7.45	7	6.725
Machine j % operating times							
Machine M_1	38.8	50	38.56	39.43	41.28	43.78	45.73
Machine M_2	63.1	81.3	62.7	64.11	67.12	71.18	74.36
Machine M_3	51.7	66.67	51.38	52.53	55	58.33	60.93
Machine M_4	25.24	32.52	25.08	25.64	26.84	28.46	29.73
Machine M_5	37.85	48.78	37.61	38.45	40.26	42.7	44.6
Machine M_6	51.7	66.67	51.38	52.53	55	58.33	60.93
Machine M_7	20.2	29.27	17.55	21.79	22.82	25.62	26.76
Machine M_8	15.2	17.89	16.3	14.74	15.44	15.66	16.54
Maximum number of parts found in internal buffer i							
Buffer 3 (p_{3-14})	6	6	6	8	12	8	8
Buffer 4 (p_{13-20})	2	2	2	2	2	3	2
Buffer 5 (p_{12-26})	1	1	1	1	1	2	1
Buffer 6 (p_{9-27})	1	1	1	1	1	2	3
Buffer 7 (p_{17-33})	6	6	6	6	6	1	2
Buffer 8 (p_{23-39})	1	1	1	1	1	3	2
Buffer 9 (p_{30-40})	1	1	1	1	1	1	1
Machine j % operational times							
Machine M_1	84.85	100	84.85	83.65	84.56	84	81.41
Machine M_2	68.87	100	68.87	68.91	75.17	71.17	69.52
Machine M_3	64.04	100	64.04	64.1	72.48	59.07	62.83
Machine M_4	57.41	100	57.41	58.33	52.68	57.65	55.39
Machine M_5	57.09	100	57.09	53.53	54.02	58.36	52.79
Machine M_6	82.02	100	82.02	81.09	76.85	78.65	76.21
Machine M_7	86.12	100	87.46	87.18	85.57	85.05	88.1
Machine M_8	85.49	100	85.58	84.62	84.56	81.49	82.16
Mean number of parts in internal buffer i							
Buffer 3 (p_{3-14})	3.8	3.5	3.88	5.02	5.4	5.85	5.4
Buffer 4 (p_{13-20})	1.25	1.32	1.25	0.25	1.25	1.49	1.32
Buffer 5 (p_{12-26})	0.25	0.32	0.25	0.25	0.25	0.49	0.32
Buffer 6 (p_{9-27})	0.42	0.48	0.25	0.52	0.45	0.7	1.16
Buffer 7 (p_{17-33})	3.4	3.59	3.25	0.25	2.83	0.63	0.74
Buffer 8 (p_{23-39})	0.54	0.67	3.25	0.57	0.58	1.77	0.88
Buffer 9 (p_{30-40})	0.22	0.25	0.2	0.21	0.58	0.25	0.25

found concurrently in each buffer. The total duration of this new simulation is 298 time units and it is completed after 775 steps. The mean production times for *type 1*and *2* final products are 7.26 and 7.45 time units respectively, which means that the finite buffer capacities used in the previous simulation are very close to optimal for the parameters of the system used here and other ways of it's optimisation should be find (the difference between the results of the simulation of Fig. 16 and the one with infinite internal buffers is about 5%). In this simulation, in parallel machines

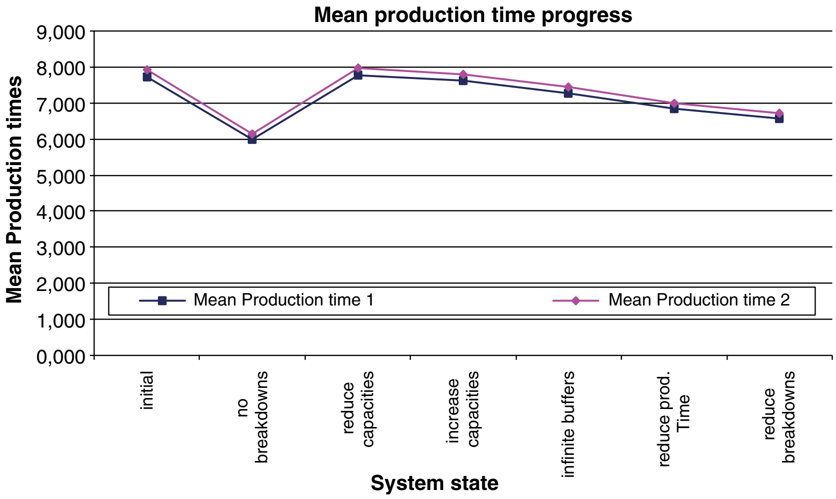

Fig. 20. Progress of the mean production cycle time of the overall system in relation with the described system changes.

module machine M_7 produces 17 parts while M_8 the remaining 23. Another interesting point is the maximum numbers of in-process parts found concurrently in the internal buffers during the simulation. This is 12 parts in buffer 3, 2 in buffer 4, 1 in buffer 5, 1 in buffer 6, 6 in buffer 7, 1 in buffer 8 and 1 in buffer 9 respectively. This shows that only for buffer 3 a further increase of capacity is useful. Buffer levels during infinite capacities simulation are shown in Fig. 17 that follows.

During the simulations it has been observed that machine M_2 is in an important way responsible for the delay of the whole system. This happens because M_2 is much slower than M_1, and this has a negative impact in the final assembly that takes place in M_6, as parts from both machines participate in this assembly. To improve the state of the system, a reduction of the production time of machine M_2 from 5 to 4 time units is attempted (The main goal of this simulation is to show that the reduction of the duration of a process has an impact in the whole system's performance, especially in the case of a machine that is much slower than neighboring. Internal buffers are again finite and their capacities are the same as in the penultimate simulation). This can be achieved by alterations on the machines' production features and tools used or even by replacement of the machines if this is necessary. After this change, simulation is completed in 725 steps and the total time needed to produce the products is reduced to 281 time units from 312 before. Mean production times are now 6.85 time units for *type 1* and approximately 7 time units for *type 2* products and are about 0.8 time units less than before (almost 11.5 % performance improvement). The new internal buffer levels are shown in Fig. 18.

In this simulation, in parallel machines module machine M_7 produces 18 parts while M_8 the remaining 22. The change of process delay has a strong impact also in the number of parts found in buffers, as buffers 3 and 8 presented by places p_{13-20} and p_{23-39} respectively are anymore full for time intervals while buffer p_{17-33} that in previous simulations had 6–7 parts now has 1 part maximum. In Fig. 18 it may be observed that after this change only two buffers reach their capacities during the simulation. Once again buffer 3 works at capacity for a significant period of the production time. Rest buffers never exceed two parts, except buffer 7 that has three parts. The level of buffer 7 has an important reduction as in the previous simulation this buffer had six parts at maximum.

To improve further the performance of the system, the components that appear high breakdown frequency are found and the alteration of this negative behaviour is attempted (main objective of this simulation is to show that high breakdown frequency of some machines has a significant negative impact in the whole system's performance). It must be pointed that not all the alterations of this type will have equally positive effects, so trial and error must be used to have satisfying results. The tried alterations must also be technically realizable. The simulation is repeated with alteration of the mean value of breakdown appearances at machine M_5 from 2 to 4 time units. The buffer levels for this simulation are presented in Fig. 19.

In this case the simulation is terminated after 719 steps and has a total duration of 269 time units. The mean production times for parts of *type 1*and *2* are 6.56 and 6.725 time units respectively. This means that there is a reduction of the mean production times about 4–4.5% in comparison with the previous simulation. Fig. 19 shows that only buffer 3 and 6 are full during the operational time of the system. In this simulation, in parallel machines module machine M_7 produces 18 parts while M_8 the remaining 22. This process may be repeated (trial and error) until all buffers work at their capacities and the optimal resources utilization is possible. Table 4 shows the progress of a number of quantities (machine's operational times, mean number of parts in internal buffers, machine's operating times, maximum number of parts concurrently in buffers etc.) while Fig. 20 summarizes the results obtained during the different simulations, with respect to the changes performed in net's characteristics. From Table 4 it is obvious that the changes performed do not only reduce the mean production times, but also have positive consequences in other system parameters like % operating times that have an increase about 15–20% between first and last simulation. Also, buffer 3 has the highest mean number of parts in comparison with all other buffers, which means that in this part of the net additive changes may lead to significant improvement of system's performance.

4.2. Discussion

The same analysis may be used in cases of production systems where parts are not available in the beginning of the process but they enter the system according to a given distribution. In such cases the minimization of *WIP* and cycle time are the main objectives and are implemented through the use of appropriate controllers that follow predetermined control strategies based on data received by sensors (Tsourveloudis *et al.*, 2000).

5. Conclusions

Modular PNs have been used for modeling, analysis and synthesis of random topology dedicated production networks. The approach may be applied to any configuration DEDS, with the advantage that analysis and synthesis is accomplished in terms of analysis and synthesis of the four derived PN modules. The calculation of the overall PN system nodes and the derivation of the PN model invariants are done in general without considering a specific topology system. Compared to other PN based approaches the proposed one is rather complete and general covering the aspects of system decomposition/composition, constraint satisfaction, complexity and performance evaluation with minimum initial assumptions and regardless of system topology. After extensive simulations and after evaluating the results, it has been observed that although the presented methodology is drastically different compared to the fuzzy logic controller presented in Tsourveloudis *et al.* (2000) and Ioannidis *et al.* (2002), the results encourage us to try to connect the two approaches: minimization of *WIP* leads to specific buffer levels, a phenomenon also captured by the derived modular PN system model. The same holds when considering machine blocking and/or starvation. This postulated observation is an additional justification of the generality and applicability of the proposed method.

Acknowledgments

This work has been partially supported from "Herokleitos" fellowships for research from the Technical University of Crete EPEAEKII-88727 and by a research grant from the Greek Secretariat for Research and Technology/European Union, E.P.A.N. M.4.3.6.1, 2000T 01330005.

Notes

1. The terms production and manufacturing systems are used interchangeably from now on.
2. The terms system and network will be used interchangeably.

3. The terms production line and transfer chain will be used interchangeably from now on.
4. Although beyond the scope of this paper, it is stated that we are currently working in integrating both methodologies, where the PN model based simulations will serve as "learning" to tune and modify the distributed fuzzy logic controller membership functions. When this is the case, system constraints are reflected in the PN model *P-invariants*. The PN model is basically the "graphical tool" while the fuzzy logic controllers provide the mathematical background.
5. Although the importance of such equipment is high, it does not consist the main objective of this work. It is assumed, that the time spent for transfer of products between machines is negligible compared to the time spent for processes and also in any moment such facilities are available to perform transferring operations with minimum delay.

References

Allam, M. and Alla, H. (1998). Modeling and simulation of an electronic component manufacturing system using hybrid Petri Nets. *IEEE Transactions on Semiconductor Manufacturing*, **11**(3), 374–383.

Akella, R. and Kumar, P. R. (1986). Optimal control of production rate in a failure prone manufacturing system. *IEEE Transactions on Automatic Control*, **31**, 116–126.

Anschuetz, Henryk (2001) HPSIM, http://home.t-online.de/home/henryk.a/petrinet/e/hpsim_e.htm.

Balduzzi, F., Giua, A. and Menga, G. (2000). First – Order Hybrid Petri Nets: A model for Optimization and Control. *IEEE Transactions on Robotics and Automation*, **16**(4), 382–399.

Bai, S. X. and Gershwin, S. B. (1990) Scheduling manufacturing systems with work-in-process inventory control: formulation of single part type systems. in Proceedings of the 29th IEEE CDC, pp 557–564.

Bai, S. X. and Gershwin, S. B. (1994). Scheduling manufacturing systems with work-in-process inventory control: multiple-part-type systems. *International Journal of Production Research*, **32**, 365–386.

Boel, R. and Stremersch, G. (1999) VHS Case Study 5: Semantics of Timed Petri nets, equivalence and model reduction, Technical Report for VHS (Verification of Hybrid Systems), Systems Group University Ghent.

Buzacott, J. A. and Shanthikumar, J. G. (1993) *Stochastic Models of Manufacturing Systems*, Englewood Cliffs, Prentice Hall, NJ.

Christensen, S. and Petrucci, L. (2000). Modular Analysis of Petri Nets. *The Computer Journal*, **43**(3), 224–242.

Conway, R., Maxwell, W., McClain, J. O. and Joseph, Thomas L. (1988). The role of work-in-process inventory in serial production lines. *Operations Research*, **36**, 229–241.

Custodio, L., Senteiro, J. and Bispo, C. (1994). Production Planning and Scheduling using a fuzzy decision system. *IEEE Transactions on Robotics and Automation*, **10**, 160–168.

David, R. and Alla, H. (1992) *Petri Nets & Grafcet – Tools for Modeling Discrete Event Systems*, Prentice Hall.

Desrochers, A. A. and Al–Jaar, R. (1995) *Applications of Petri Nets in Manufacturing Systems – Modeling, Control and Performance Analysis*, IEEE Press.

Domg, M. and Chen, F. F. (2001). Process modelling and analysis of manufacturing supply chain networks using object-oriented Petri nets. *Robotics and Computer Integrated Manufacturing*, **17**, 121–129.

Feldmann, K., Schnur, C. and Colombo, W. (1996). Modularised, distributed real-time control of flexible production cells, using petri nets. *Control Engineering Practice*, **4**(8), 1067–1078.

Frey, G. (2000). Assembly line sequencing based on Petri-net T-invariants. *Control Engineering Practice*, **8**, 63–69.

Gershwin, S. B. (1994) *Manufacturing Systems Engineering*, Prentice Hall, New Jersey.

Gershwin, S. B. (2000). Design and operation of manufacturing systems: the control-point policy. *IIE Transactions*, **32**, 891–906.

Gu, T. and Bahri, P. (2002). A survey of Petri net applications in batch processes. *Computers in Industry*, **47**, 99–111.

Ioannidis, S., Tsourveloudis, N. and Valavanis, K. (2002) Supervisory control of multiple-part-type production networks, CD-ROM Proceedings, 10th IEEE Mediterranean Conference on Control and Automation, Lisbon, Portugal.

Jehng, Wern-Kueir. (2002). Petri net models applied to analyze automatic sequential pressing syst *ems. Journal of Materials Processing Technology*, **120**, 115–125.

Jeng, M. D. and Xie, X. (2001). Modeling and analysis of Semiconductor Manufacturing Systems with Degraded Behaviour Using Petri Nets and Siphons. *IEEE Transactions on Robotics and Automation*, **17**(5).

Martinez, J. and Silva, M. (1982). A Simple and fast algorithm to obtain all invariants of generalized Petri net. *Application and Theory of Petri Nets*, **52**, 301–311.

Mellado, E. L. (2002). Analysis of discrete event systems by simulation of timed Petri net models. *Mathematics and Computers in Simulation*, **61**, 53–59.

Moody, J. and Antsaklis, P. (1998) *Supervisory Control of Discrete Event Systems Using Petri Nets*, Kluwer Academic Publications.

Murata, T. (1989). Petri Nets: properties, analysis and applications. *Proceedings of the IEEE*, **77**(4), 541–580.

Nketsa, A. and Valette, R. (2001). Rapid and modular prototyping-based Petri nets and distributed simulation for manufacturing systems. *Applied Mathematics and Computation*, **120**, 265–278.

Peterson, J. (1981) *Petri Net Theory And The Modeling of Systems*, Prentice Hall Inc.

Proth, J.M. and Sauer, N. (1998). Scheduling of piecewise constant product flows: a Petri net approach. *European Journal of Operational Research*, **106**, 45–56.

Proth, J.M. and Xiaolan, X. (1996) *Petri Nets, A Tool for Design and Management of Manufacturing Systems*, John Wiley & Sons.

Reisig, W. (1992) *A Primer in Petri Net Design*, Springer Verlag, Germany.

Rogers, G. and Bottaci, L. (1997). Modular production systems: a new manufacturing paradigm. *Journal of Intelligent Manufacturing*, **8**, 147–156.

Tang, Y., Zhou, M. C. and Caudill, R. (2001). An integrated approach to disassembly planning and demanufacturing operation. *IEEE Transactions on Robotics and Automation*, **17**(6), 773–784.

Thevenon, L. and Flaus, J. M. (2000). Modular representation of complex hybrid systems: application to the simulation of batch processes. *Simulation Practice and Theory*, **8**(5), 283–306.

Thomas, J., Nissanke, N. and Baker, K. (1996). A hierarchical Petri Net framework for the representation and analysis of assembly. *IEEE Transactions on Robotics and Automation*, **12**(2), 248–279.

Torn, A. Petri nets – Calculating invariants, online at www.abo.fi/~atorn/Petri/P53.html.

Tsinarakis, G. (2002) Modeling Synthesis and Analysis of Production Systems with use of Petri Nets Master Thesis (in Greek), Technical University of Crete, Department of Production Engineering and Management.

Tsourveloudis, N., Dretoulakis, E. and Ioannidis, S. (2000). Fuzzy work-in-process inventory control of unreliable manufacturing systems. *Information Sciences*, **27**, 69–83.

Valavanis, K. (1990). On the Hierarchical Modeling Analysis and Simulation of Flexible Manufacturing Systems with Extended Petri Nets. *IEEE Transactions on Systems, Man and Cybernetics*, **20**(1), 23–23.

Valette, R. and Chezalviel-Pradin, B. (1999) An Introduction to Petri net theory Fuzziness in Petri nets. in *Studies in fuzziness and soft computing*, J. Cardoso and H. Carmago. Physica Verlag, pp. 3–24.

Veltsos, C. (1998) Automated Petri Net Synthesis for the Modeling and Analysis of Complex Dynamic Systems, Dissertation presented to the Graduate Faculty of The University of Southwestern Louisiana.

Venkatesh, K. and Ilyas, M. (1995). Real-time Petri nets for modelling controlling and simulation of local area networks in flexible manufacturing systems. *Computers in Engineering*, **28**(2), 147–162.

Wang, L. C. (1996). Object-oriented Petri nets for modeling and analysis of automated manufacturing systems. *Computer Integrated Manufacturing Systems*, **26**(2), 111–125.

Wang, L. C. and Wu, S. Y. (1998). Modeling with colored timed object-oriented petri nets for automated manufacturing systems. *Computers in Engineering*, **34**(2), 463–480.

Wein, L. (1988). Scheduling semiconductor wafer fabrication. *IEEE Transactions on Semiconductor Manufacturing*, **1**, 115–130.

Xue, Y., Kieckhafer, R. M. and Choobineh, F. F. (1998). Automated construction of GSPN models for flexible manufacturing systems. *Computers in Industry*, **37**, 17–25.

Zhou, M. C. and DiCesare, F. (1993) *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*, Kluwer Academic Publishers.

Zuberek, W. (1998) TPN-tools – tools for modeling and performance analysis using Timed Petri nets, online at web.cs.mun.ca/~wlodek/tpntools/tpntools-invar.html.

Zuberek, W. (2001). Timed Petri nets in modeling and analysis of cluster tools. *IEEE Transactions on Robotics and Automation*, **17**(5), 562–575.

Zuberek, W. and Kubiak, W. (1999). Timed Petri Nets in Modeling and Analysis of Simple Schedules for Manufacturing Cells. *Computers and Mathematics with Applications*, **37**, 191–206.