

## **Decizie de indexare a faptei de plagiat la poziția 00420 / 06.02.2018 și pentru admitere la publicare în volum tipărit**

**care se bazează pe:**

**A. Nota de constatare și confirmare a indiciilor de plagiat prin fișa suspiciunii inclusă în decizie.**

Fișa suspiciunii de plagiat / Sheet of plagiarism's suspicion	
Opera suspicionată (OS) Suspicious work	Opera autentică (OA) Authentic work
OS	CIUFUDEAN, Călin, GRAUR, Adrian., FILOTE, Constantin and TURCU, Cornel. A New Formalism for Failure Diagnosis: Ant Colony Decision Petri Nets. <i>JSW</i> , 2(1), 2007. pp.39-46.
OA	CIUFUDEAN, Călin; GRAUR, Adrian; FILOTE, Constantin, TURCU, Cornel and POPA, Valentin. Diagnosis of complex systems using ant colony decision Petri nets. <i>Availability, Reliability and Security, ARES 2006. The First International Conference on. IEEE</i> , 2006. pp.8-15.
Incidența minimă a suspiciunii / Minimum incidence of suspicion	
P01:	p.40:01s – p.40:10d
P02:	p.41:Fig.1
P03:	p.42:01d – p.43:00s
Fișa întocmită pentru includerea suspiciunii în Indexul Operelor Plagiate în România de la Sheet drawn up for including the suspicion in the Index of Plagiarized Works in Romania at <a href="http://www.plagiate.ro">www.plagiate.ro</a>	

**Notă:** Prin „p.72:00” se înțelege paragraful care se termină la finele pag.72. Notăția „p.00:00” semnifică până la ultima pagină a capitolului curent, în întregime de la punctul inițial al preluării.

**Note:** By „p.72:00” one understands the text ending with the end of the page 72. By „p.00:00” one understands the taking over from the initial point till the last page of the current chapter, entirely.

**B. Fișa de argumentare a calificării de plagiat alăturată, fișă care la rândul său este parte a deciziei.**

Echipea Indexului Operelor Plagiate în România

## Fișa de argumentare a calificării

Nr. crt.	Descrierea situației care este încadrată drept plagiat	Se confirmă
1.	Preluarea identică a unor pasaje (piese de creație de tip text) dintr-o operă autentică publicată, fără precizarea întinderii și menționarea provenienței și însușirea acestora într-o lucrare ulterioară celei autentice.	✓
2.	Preluarea a unor pasaje (piese de creație de tip text) dintr-o operă autentică publicată, care sunt rezumate ale unor opere anterioare operei autentice, fără precizarea întinderii și menționarea provenienței și însușirea acestora într-o lucrare ulterioară celei autentice.	
3.	Preluarea identică a unor figuri (piese de creație de tip grafic) dintr-o operă autentică publicată, fără menționarea provenienței și însușirea acestora într-o lucrare ulterioară celei autentice.	✓
4.	Preluarea identică a unor tabele (piese de creație de tip structură de informație) dintr-o operă autentică publicată, fără menționarea provenienței și însușirea acestora într-o lucrare ulterioară celei autentice.	
5.	Republicarea unei opere anterioare publicate, prin includerea unui nou autor sau de noi autori fără contribuție explicită în lista de autori	
6.	Republicarea unei opere anterioare publicate, prin excluderea unui autor sau a unor autori din lista inițială de autori.	✓
7.	Preluarea identică de pasaje (piese de creație) dintr-o operă autentică publicată, fără precizarea întinderii și menționarea provenienței, fără nici o intervenție personală care să justifice exemplificarea sau critica prin aportul creator al autorului care preia și însușirea acestora într-o lucrare ulterioară celei autentice.	✓
8.	Preluarea identică de figuri sau reprezentări grafice (piese de creație de tip grafic) dintr-o operă autentică publicată, fără menționarea provenienței, fără nici o intervenție care să justifice exemplificarea sau critica prin aportul creator al autorului care preia și însușirea acestora într-o lucrare ulterioară celei autentice.	✓
9.	Preluarea identică de tabele (piese de creație de tip structură de informație) dintr-o operă autentică publicată, fără menționarea provenienței, fără nici o intervenție care să justifice exemplificarea sau critica prin aportul creator al autorului care preia și însușirea acestora într-o lucrare ulterioară celei autentice.	
10.	Preluarea identică a unor fragmente de demonstrație sau de deducere a unor relații matematice care nu se justifică în regăsirea unei relații matematice finale necesare aplicării efective dintr-o operă autentică publicată, fără menționarea provenienței, fără nici o intervenție care să justifice exemplificarea sau critica prin aportul creator al autorului care preia și însușirea acestora într-o lucrare ulterioară celei autentice.	
11.	Preluarea identică a textului (piese de creație de tip text) unei lucrări publicate anterior sau simultan, cu același titlu sau cu titlu similar, de un același autor / un același grup de autori în publicații sau edituri diferite.	
12.	Preluarea identică de pasaje (piese de creație de tip text) ale unui cuvânt înainte sau ale unei prefețe care se referă la două opere, diferite, publicate în două momente diferite de timp.	

### Notă:

a) Prin „proveniență” se înțelege informația din care se pot identifica cel puțin numele autorului / autorilor, titlul operei, anul apariției.

b) Plagiatul este definit prin textul legii<sup>1</sup>.

„...plagiatul – expunerea într-o operă scrisă sau o comunicare orală, inclusiv în format electronic, a unor texte, idei, demonstrații, date, ipoteze, teorii, rezultate ori metode științifice extrase din opere scrise, inclusiv în format electronic, ale altor autori, fără a menționa acest lucru și fără a face trimitere la operele originale...”.

Tehnic, plagiatul are la bază conceptul de **piesă de creație** care<sup>2</sup>:

„...este un element de comunicare prezentat în formă scrisă, ca text, imagine sau combinat, care posedă un subiect, o organizare sau o construcție logică și de argumentare care presupune niște premise, un raționament și o concluzie. Piesa de creație presupune în mod necesar o formă de exprimare specifică unei persoane. Piesa de creație se poate asocia cu întreaga operă autentică sau cu o parte a acesteia...”

cu care se poate face identificarea operei plagiate sau suspicioase de plagiat<sup>3</sup>:

„...O operă de creație se găsește în poziția de operă plagiată sau operă suspicioasă de plagiat în raport cu o altă operă considerată autentică dacă:

- i) Cele două opere tratează același subiect sau subiecte înrudite.
- ii) Opera autentică a fost făcută publică anterior operei suspicioase.
- iii) Cele două opere conțin piese de creație identificabile comune care posedă, fiecare în parte, un subiect și o formă de prezentare bine definită.
- iv) Pentru piesele de creație comune, adică prezente în opera autentică și în opera suspicioasă, nu există o menționare explicită a provenienței. Menționarea provenienței se face printr-o citare care permite identificarea piesei de creație preluate din opera autentică.
- v) Simpla menționare a titlului unei opere autentice într-un capitol de bibliografie sau similar acestuia fără delimitarea întinderii preluării nu este de natură să evite punerea în discuție a suspiciunii de plagiat.
- vi) Piesele de creație preluate din opera autentică se utilizează la construcții realizate prin juxtapunere fără ca acestea să fie tratate de autorul operei suspicioase prin poziția sa explicită.
- vii) În opera suspicioasă se identifică un fir sau mai multe fire logice de argumentare și tratare care leagă aceleași premise cu aceleași concluzii ca în opera autentică...”

<sup>1</sup> Legea nr. 206/2004 privind buna conduită în cercetarea științifică, dezvoltarea tehnologică și inovare, publicată în Monitorul Oficial al României, Partea I, nr. 505 din 4 iunie 2004

<sup>2</sup> ISOC, D. Ghid de acțiune împotriva plagiatului: bună-conduită, prevenire, combatere. Cluj-Napoca: Ecou Transilvan, 2012.

<sup>3</sup> ISOC, D. Prevenitor de plagiat. Cluj-Napoca: Ecou Transilvan, 2014.

# A New Formalism for Failure Diagnosis: Ant Colony Decision Petri Nets

Calin Ciufudean, Adrian Graur, Constantin Filote, and Cornel Turcu

“Stefan cel Mare” University /Faculty of Electrical Engineering and Computer Science, Suceava, Romania  
Email: {calin, adriang, filote, cturcu}@eed.usv.ro

**Abstract**— Failure diagnosis in large and complex systems is a critical task. A discrete event system (DES) approach to the problem of failure diagnosis is presented in this paper. A classic solution to solve DES’s diagnosis is a stochastic Petri net. The foraging behavior of ant colonies can give rise to the shortest path, which will reduce the state explosion of stochastic Petri net. Therefore, a new model of stochastic Petri net, based on foraging behavior of real ant colonies is introduced in this paper. This model can contribute to the diagnosis, the performance analysis and design of supervisory control systems.

**Index Terms**— Stochastic Petri nets, discrete-event systems, Ant Colony Optimization algorithm, diagnosis of complex systems.

## I. INTRODUCTION

Diagnosis is a crucial and challenging task in the automatic control of complex systems, e.g., in flexible manufacturing systems. In this paper a discrete event system (DES) approach to the problem of diagnosis of complex systems is presented. The property of diagnosability is introduced in the context of the failure diagnosis problem, e.g., in the context of the availability of the DES. We propose a systematic procedure for diagnosis implemented with a new class of stochastic Petri nets (GSPN’s); i.e., ant colony decision Petri nets (ADPN) [1] and related models (e.g., stochastic reward nets [2] stochastic activity networks [3]) are gaining increased acceptance as tools for analyzing complex systems. The acceptance of such high-level formalism is due to their ability to represent complex systems in a compact and convenient way, while still describing an underlying continuous-time Markov chain (CTMC) [4].

This method suffers from the well-known state explosion problem: a GSPN can determine an underlying CTMC with a large number of states. This problem severely limits the size of models for which an exact analysis can reasonably be attempted. This problem has received considerable attention in the literature, and a

wide variety of algorithms have been proposed. Stochastic Petri nets (SPN) were developed by associating transitions/places with exponentially distributed random time delays [5], [6]. Generalized SPN [7], [8] allowed the inclusion of immediate transition and inhibitor arcs. Analysis tools have been developed [9], [10]. These methods are all based on results obtained from the underlying Markov chain for such system models. Extended SPN (ESPN) [11] were developed to allow generally distributed, including deterministic, transition delays for non-concurrent transitions. The underlying models of these PN are semi/Markov processes. In [12] Sampath et al. proposed a diagnosis approach for discrete event systems. They introduced the notion of diagnosability and gave a necessary and sufficient condition to test it. Their condition is expressed as a property of the diagnoser of the system. In order to test the diagnosability, the diagnoser needs to be constructed first. The complexity of constructing the diagnoser and testing the diagnosability is exponential in the number of states of the system and double exponential in the number of failure types. Ant Colony Optimisation (ACO) is a recently developed approach that takes inspiration from the behavior of real ant colonies to solve NP - hard optimisation problems. The ACO meta-heuristic was first introduced by Dorigo [13], and was defined by Dorigo, Di Caro and Gambardella [14]. It has been successfully applied to various hard combinatorial optimization problems. In this paper we present the first application of ACO to Petri nets formalism, in order to simplify the models achieved with GSPN for solving the diagnosis of complex systems. In section 2 we briefly introduce the Ant Colony Optimization algorithm. Then we describe the structure of our diagnoser in section 3. In section 4 we present the experimental results implemented on an FMS. Finally, we summarize our findings and conclude with some discussion.

## II. ANT COLONY OPTIMIZATION ALGORITHMS

A Bayesian network (BN) is a directed acyclic graph where nodes represent random variables and edges represent conditional dependencies between random variables. Attached to each node there is a conditional probability table (CPT) that describes the conditional probability distribution of that node given its parents’

Based on “Diagnosis of Complex Systems Using Ant Colony Decision Petri Nets”, by C. Ciufudean, A. Graur, C. Filote, C. Turcu, and V. Popa, which appeared in the Proceedings of the IEEE International Conference on ARES 2006, Vienna, Austria, April 2006. © 2006 IEEE.

**P01** states [15]. Although the distributions in a BN can be discrete or continuous, we shall consider discrete ones. Search algorithms have been studied extensively in combinatorial optimization. Researches have applied various search strategies, for example, the best first search [16], linear programming [17], stochastic local search [18], genetic algorithms [19], etc. Ant algorithms were inspired by the foraging behavior of real ant colonies, i.e., how ants can find the shortest path between food sources and nest. Ants deposit on the ground a chemical substance called pheromone while walking. This forms pheromone trails through which ants can find the way, and also provides indirect communication among ants. It has been shown experimentally [13] that this foraging behavior can give rise to the emergence of the shortest path when employed by a colony of ants. Based on this ant colony foraging behavior, ACO algorithms using artificial ant systems to solve hard discrete optimization problems have been developed. In an ant system, artificial ants are created to explore the search space simulating real ants searching their environment. The objective values to be optimized usually correspond to the quality of the food and the length of the path to the food. The artificial ants can make use of some local heuristic functions to help choose among a set of feasible solutions. In an ant system, artificial ants build solutions by moving on the Bayesian network from one node to another. When an ant visits node  $x_i$ , it must take a conditional branch which is a number in the CPT. For evidence nodes  $A$ , ants are only allowed to take the branches that agree with  $A$ . Each node in BN has three tables: the Pheromone Table (PT), the Heuristic Function Table (HFT), and the Ant Decision Table (ADT). The PTs store pheromone values accumulated on each conditional branch. HFTs represent heuristics used by ants. ADTs are used by ants to make the final decision of choosing which branch to take. The ADT,  $A_i = [a_{ijk}]$ , of node  $x_i$  is obtained by the composition of the local pheromone trail values  $ph_{ijk}$  with the local heuristic values  $h_{ijk}$  as follows [14]:

$$a_{ijk} = \frac{(ph_{ijk})^\alpha \cdot (h_{ijk})^\beta}{\sum_j (ph_{ijk})^\alpha \cdot (h_{ijk})^\beta} \quad (1)$$

where  $j$  is the  $j^{\text{th}}$  row and  $k$  is the  $k^{\text{th}}$  column of the corresponding ADT at the  $i^{\text{th}}$  node. Parameters  $\alpha$  and  $\beta$  control the relative weight of pheromone trails and heuristic values.

We also know [13], [14] the probability with which an ant chooses to take a certain conditional branch:

$$p_{ij} = \frac{a_{ij\pi_i}}{\sum_j a_{ij\pi_i}} \quad (2)$$

where  $\pi_i$  is the column index of the ADT and its value is conditioned on the values of parent nodes of  $i^{\text{th}}$  node.

After ants have built their tour (a diagnosis), each ant deposits pheromone  $\Delta ph_{ijk}$  on the corresponding pheromone trails (i.e., the conditioned branches of each

node of the tour). For us, the pheromone value represents the probability to cover the selected tour (e.g., by anticipation of the next section, we show that the pheromone value  $i$  represents the probability of firing transition  $i$  in the SPN), as follows:

$$\Delta ph_{ijk} = \begin{cases} P(x_1, \dots, x_n), & j = x_i, k = \pi(x_i) \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where  $P(x_1, \dots, x_n)$ , is:

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i / \pi(x_i)) \quad (4)$$

Where,  $\pi(x_i)$  denotes the parent nodes of  $x_i$ .

Each ant drops pheromone to one cell of each PT at each node, i.e., the  $j^{\text{th}}$  row,  $k^{\text{th}}$  column of the PT at  $i^{\text{th}}$  node. After dropping the pheromone, the ant dies.

### III. THE ANT COLONY DECISION PETRI NET DIAGNOSER

The complex system, e.g., a flexible manufacturing system (FMS) to be diagnosed is modelled as a finite state machine of DES's formalism:

$$W = (S, E, t, m_0) \quad (5)$$

Where  $S$  is the state space,  $E$  is the set of events,  $t$  is the partial transition function and  $m_0$  is the initial state of the system. The model  $W$  accounts for the normal and failed behaviour of the system. Let  $E_f \subseteq E$  denote the set of failure events which are to be diagnosed. Our objective is to identify the occurrence of the failure events. Therefore we partition the set of failure events into disjoint sets corresponding to different failure types:

$$E_f = E_{f_1} \cup E_{f_2} \cup \dots \cup E_{f_m} \quad (6)$$

This partition is motivated by the following considerations [5]:

- 1) Inadequate instrumentation may render it impossible to diagnose uniquely every possible fault;
- 2) It may not be required to identify uniquely the occurrence of every failure event. We may simply be interested in knowing whether failure event has happened as the effect of the same failures in the system.

So, when we say that "a failure of type  $F_i$  has occurred", we mean that some event from the set  $E_{f_i}$  has occurred.

In [5],[6] the diagnosability is defined as follows: A prefix-closed and live language  $L$  is said to be I-diagnosable with respect to the projection  $P$ , the partition  $E_f$ , and the indicator  $I$  if the following holds:

$$\left( \forall i \in E_f \right) \cdot \left( \exists n \in N \right) \cdot \left( \forall s \in E_{f_i} \right) \cdot \left( \forall t \in \frac{L}{s} : st \in I(E_{f_f}) \right) \left[ \|t\| \geq n; \Rightarrow D \right] \quad (7)$$

Where the diagnosability condition D is:

$$\omega \in P_L^{-1}[P(st)] \Rightarrow E_{f_i} \in \omega \tag{8}$$

Note that  $I(E_{fi})$  denotes the set of all traces of L that end in an event from the set  $E_{fi}$ . The behaviour of the system is described by the prefix-closed live language L (A) generated by A (see relation (1)). L is a subset of  $E^*$ , where  $E^*$  denotes the kleen closure of the set E [7].  $\|s\|$  denotes the length of trace  $s \in E$ .  $L/s$  denote the post language of L after s, i.e.

$$\frac{L}{s} = \{t \in E^* / st \in L\} \tag{9}$$

We define the projection  $P: E^* \rightarrow E$  in the usual manner [8]:

$$P(\varepsilon) = \varepsilon \text{ and } P(s_1s_2) = P(s_1) \cdot P(s_2), \\ s_1 \in E^* \text{ and } s_2 \in E \tag{10}$$

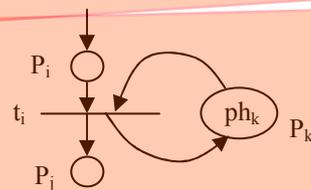
Where  $\varepsilon$  denotes the empty trace.

The above definition, e.g. relations (7) and (8), means the following: Let s be any trace generated by the system that ends in a failure event from the set  $E_{fi}$ , and let t be any sufficiently long continuation of s.

Condition D then requires that every trace belonging to the language that produces the same record of observable events, and in which the failure event is followed by certain indicator, should contain a failure event from the set  $E_{fi}$ . This implies that on some continuation of s one can detect the occurrence of a failure of the type  $F_i$  with a finite delay, specifically in at most  $n_i$  transitions of the system after s. To summarize, here diagnosability requires detection of failures only after the occurrence of an indicator event corresponding to the failure. In this paper we improve this approach by according a gradual importance of failure indicators, in correspondence with the availability of the system. In our assumption the diagnoser is a stochastic Petri net (SPN), where the places are marked with the availability of the correspondent production cell. The availability of a production cell is calculated with a Markov chain, where the transitions reflect the gradual importance of the failures in the cell. We may say that the diagnoser is an extended observer where we append a label to every estimated state. The labels carry failure information and failures are diagnosed by checking these labels. We also assume the system W is normal at the start.

A diagnoser is a deterministic finite state machine whose transitions correspond to observations and whose states correspond to the set of system states and failures that are consistent with the observations. The transitions of the diagnoser are labelled with observable events, and the states of the diagnoser are labelled with sets of pairs  $(v, l)$  denoting a state and a failure label of the abstracted model. In our approach, the diagnoser efficiently maps observations to sets of possible system states and failures,

and it is modelled with a new class of Petri nets, called here Ant Colony Decision Petri Nets (ADPN), which are an extension of our previous work [20] where we introduced the Stochastic Coloured Petri Nets (SCP). Here, the colour of tokens in ADPN, represents the colour of the ants, grouped in families. We suppose that in our model there are different ant families (e.g., red ants, black ants, s.a.), each kind of ant has a specific pheromone; an ant will sense the pheromone in the nodes of the net and will follow only the specific path that was marked with the pheromone of its family. In the initial marking of the Petri net we know the number of the test ants, by colour. Considering that after firing a transition in the net, the ant leaves its pheromone in the control place of the respective transition (see fig. 1), and then dies, after the first ant reaches the end of the graph we count the number of the ants remained in the first place of the net. We conclude which is the shortest way in the net i.e., which family of ants found the optimum path, considering that a family of ants will never follow the same way as another ant family.



P02

Fig. 1 The basic structure of ADPN

In Fig.1. one can see that the control place,  $p_k$ , of transition  $t_i$  memorizes the pheromone of the ant which burns first the transition  $t_i$ . We say that transition  $t_i$  will be fired only by ants with colour  $ph_k$ , where  $ph_k$  has the same signification as that given in relation (1).

The firing rates of transitions in ADPN are given by the next relation:

$$f_i = \frac{(ph_i)^{\alpha_i} \cdot (h_i)^{\beta_i}}{(ph_i)^{\alpha_i} + (h_i)^{\beta_i}} \tag{11}$$

In relation (11)  $ph_k$  is the pheromone dropped in the control place by the first ant, that burns the transition  $t_i$ ;  $h_i$  is the classic exponential firing rate of a transition in a stochastic Petri net; probabilities  $\alpha_i$  and  $\beta_i$  control the failure rate, respectively the repair rate of elements (machines, electronic devices, etc) of a complex system, such as a flexible manufacturing system (FMS). We define our ADPN as follows:

An ADPN is a fire-tuple  $(P, T, k, m, V)$ , where:

$P = \{p_1, p_2, \dots, p_n\}$ ,  $n > 0$ , and is a finite set of places;

$T = \{t_1, t_2, \dots, t_s\}$ ,  $s > 0$ , and is a finite set of transitions with  $P \cup T \neq \emptyset$ ,  $P \cap T = \emptyset$ ;

$K = \{Pk_1, Pk_2, \dots, Pk_s\}$ ,  $s > 0$ , and is a finite set of pheromone - control places;

$m : P \rightarrow \mathbb{N}$ , and is a marking whose  $i^{th}$  component is the number of tokens in the  $i^{th}$  place. An initial marking is denoted by  $m_0$ ;

$V : T \rightarrow R$ , is a vector whose component is a firing time delay with an ant decision function.

In our work we assumed that when a device, sensor, transducer or any other hardware component of the analyzed system, (e.g., a FMS) fails, the system reconfiguration (after repairing it) is often less than perfect. The notion of imperfection is called imperfect coverage, and is defined as probability  $c$  that the system successfully reconfigures given that component fault occurs. The imperfect repair of a component implies that when the repair of the failed component is completed it is not "as good as new". A dependability model for diagnosability of flexible manufacturing systems is presented. The meaning of dependability here is twofold:

- System diagnosability and availability
- Dependence of the performance of the FMS on the performance of its individual physical subsystems and components.

The model considers the task-based availability of an FMS, where the system is considered operational as long as its task requirements are satisfied; respectively the system throughput exceeds a given lower bound. We model the FMS with ADPN. We decompose the FMS in productions cells. In our assumption the availability of a cell  $j$  ( $j=1,2,\dots,n$ , where  $n$  is the total number of part type cells in the FMS) is calculated with a Markov chain which includes the failure rates, repair rates, and coverability of the respective devices in the production cell  $i$ . The colour domains of transitions that load cell  $i$  include colours that result in a value between 0 and 1; the biggest value designates the cell (respectively the place in the ADPN model) which ensures the liveness of the net, respectively which will validate and burn its output transition. We assume that the reader is familiar with Petri nets theory and their applications to manufacturing systems or we refer the reader to [6], [7]. Each part entering the system is represented by a token. The colour of the token associated with a part has two components [8]. The first component is the part identification number and the second component represents the set of possible next operations determined by the process plan of the part. It is the second component that is recognized by the stochastic colours Petri net model, and the first component is used for part tracking and reference purposes. Let  $B_i$  be a  $(1 \times m)$  binary vector representing all the operations needed for the complete processing of part type  $i$ . Let  $E_i$  be a  $(m \times m)$  matrix representing the precedence relations among the operations of part type  $i$ , where  $m$  is the number of operations that are performed in the respective cell  $j$  ( $j=1,2,\dots, n$ ). For a part to be processed in the cell  $j$  it requires at least one operation that can be performed in the cell, that implies  $B_j > 0$ . Also, for a part type where there is no precedent relationship between required operations,  $E_i$  is a matrix of zeros.

For a part with identification  $x$  and part type  $y$ , the initial colour of the corresponding token is:

$$V_{yx} = [yx, B_y - (B_y \cdot E_y)] \quad (12)$$

Where  $(B_y \cdot E_y)$  is a matrix of multiplication.

For example consider the process plan of part type  $L_1$  and  $L_2$  shown in Fig.2.

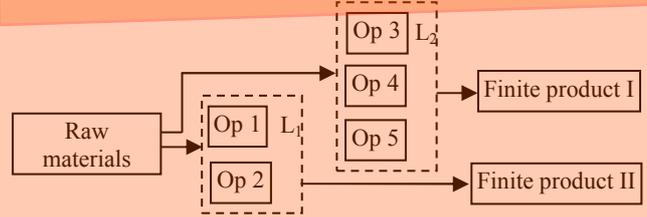


Fig.2. Process plan of part type  $L_1$  and  $L_2$

Our process plan first requires operation  $op1$  and then operation  $op2$  for complete processing. We assume that our FMS can complete 5 different types of operations (e.g., for simplicity we consider only 5 different types of operations). For part type  $L_1$ , we have:  $B_{L1} = [00011]$ .

P03

$$E_{L1} = \begin{pmatrix} & op5 & op4 & op3 & op2 & op1 \\ op5 & 0 & 0 & 0 & 0 & 0 \\ op4 & 0 & 0 & 0 & 0 & 0 \\ op3 & 0 & 0 & 0 & 0 & 0 \\ op2 & 0 & 0 & 0 & 0 & A_2 \\ op1 & 0 & 0 & 0 & A_1 & 0 \end{pmatrix}$$

Where  $A_1$  is the availability of production cell 1 (which performs operation 1), and  $A_2$  represents the availability of production cell 2 at time  $t$ . The availability  $A_i$  of cell  $i$  is calculated, as shown below, with Markov chains. We notice that  $A_i$  is re-evaluated at each major change in the process plan of FMS (such as occurrence of events: damages of hardware equipments, changes of process plan, etc). Assuming that  $A_1 > A_2$ , then we assign to  $A_1$  value 1 and to  $A_2$  value 0, so that applying relation (12), the initial color of the token corresponding to a part that belongs to part type  $L_1$  with identification mark 1, would be  $V_{L1,1} = (L_{1,1}, 00001)$ . Note that the information carried by the color of the tokens in the SCPN indicates the next operation to be performed by the FMS. Generally, we may say that  $V$  is the set of colors that represent all the possible combinations of operations that can be performed in the FMS. Each member of the set  $V$  is a vector with  $m$  components, where  $m$  is the maximum number of operations to be performed in the cells of the FMS. For example, in an FMS with 5 operations to be performed, we may have  $V = \{00000, 00001, \dots, 11111\}$ . For simplicity, we assume that operations in FMS are mapped to places in the SCPN model, places which are labeled with the operation identification number. The requirement for a production cell  $j$  ( $j=1, \dots, n$ ) which have  $N_i$  ( $i=1, \dots, m$ ) devices of type  $i$ , is that at least  $k_i$  of these devices must be operational for the FMS to be operational. To determine the system availability which includes imperfect coverage and repair, a failure state due to imperfect coverage and repair was introduced [4]. To explain the impact of imperfect coverage, we consider the

system given in Fig.3 which includes two identical manufacturing devices  $M_1$  and  $M_2$ .

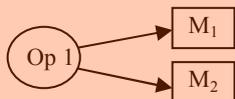


Fig.3. Example of operation performed by two identical devices

If the coverage of the system is perfect, i.e.  $c=1$ , then operation op1 is performed as long as one of the devices is operational. If the coverage is imperfect, then operation op 1 fails with probability  $1-c$ , if one of the devices  $M_1$  or  $M_2$  fails. We may say that, if operation op 1 has been scheduled on device  $M_1$  that has failed, then the system in Fig.3 fails with probability  $1-c$ . The Markov chain for manufacturing cell  $j$  is shown in Fig.4. In Fig.4 the parameters  $\lambda$ ,  $\mu$ ,  $c$ ,  $r$  denote respectively the failure rate, repair rate, coverage factor and the successful failure repair rate of devices in the cell. The first part of the horizontal transition rate with the term  $1-c$  represents the failure due to imperfect coverage of an alternative equipment. The second part, with the term  $1-r$  represents imprecise repair of the devices.

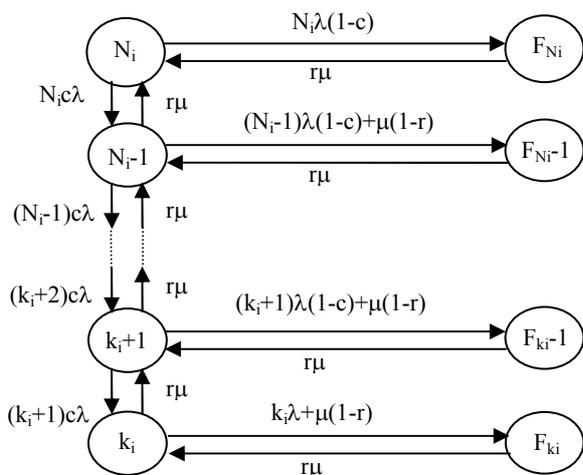


Fig.4. Markov model for cell i

The vertical transitions reflect the failure and repair of the equipments. We assume that only one device fails at a time, in a certain operation cell. At state  $N_i$  cell  $i$  is functioning with all  $N_i$  devices operational. At state  $k_i$  there are only  $k_i$  devices operational. The state of cell  $i$  changes from working state  $w_i$ , for  $k_i \leq w_i \leq N_i$ , where  $w_i$  is the number of operational devices at a certain moment, to failed state  $F_i$ , either due to imperfect coverage  $(1-c)$  or due to imperfect repair  $(1-r)$ . If the fault coverage of the system and repair of the components are perfect, the Markov chain in Fig.4 reduces to one-dimension model. The solution of the Markov chain model given in Fig.4 is a probability that at least  $k_i$  devices are working at time  $t$ . The availability of cell  $i$  is given by the next relation [5]:

$$A_i(t) = \sum_{w_i=k_i}^{N_i} P_{ki}(t), \text{ for } i=1,2,\dots,n \quad (13)$$

Where  $A_i(t)$ =the availability of cell  $i$  at moment  $t$ ;  
 $P_{ki}(t)$ =probability of  $k_i$  devices being operational in cell  $i$  at time  $t$ ;

$N_i$ =total number of devices of type  $j$  in cell  $i$ ;  
 $K_i$ =required minimum number of operational devices in cell  $i$ .

After a Markov chain for each cell of the measuring system is constructed and desired probabilities  $A_i(t)$ ,  $i=1,2,\dots,n$  corresponding to each manufacturing cell are determined, the ant colony decision Petri net (ADPN) can be initialized and the simulation process of the FMS begins. The status of this graph (e.g., the ADPN) at different moments  $t_k$ , gives us the diagnosis of the FMS.

#### IV. ILLUSTRATIVE EXAMPLE

In this section, we exemplify the above presented approaches on a flexible manufacturing system. We give the relative error in aggregated measures, such as the mean number of tokens in a given place or a throughput of transitions. Markov chain was solved using Gauss-Seidel with iterations continuing until the relative element-wise difference between subsequent probability vectors was less than  $10^{-6}$ . The flexible manufacturing system consists of two cells linked together by a conveyor system. Each cell consists of a machine to handle within-cell part movement. Work-pieces enter the system at the Load/Unload station, where they are released from two buffers,  $A$  and  $B$ , and then are sorted in cells (pieces of type "a" in one cell, and pieces of type "b" in the other cell). We notice that in the buffer  $A$  there are pieces of types "a", "b", and others. In buffer  $A$  the number of pieces "a" is greater than the number of pieces "b". In the buffer  $B$ , there are pieces of types "a", "b", and others, where the number of pieces "b" is greater than the number of pieces "a". The conveyor moves pieces between the Load/Unload station and those two cells. The finished (sorted) work-piece leaves the system, and a raw work-piece (unsorted piece) enters the system, respectively in one of those two buffers  $A$  or  $B$ . The maximum number of work-pieces permitted inside a cell at any given time is limited. The conveyor along with the central storage incorporates a sufficiently large buffer space so that it can be thought of as possessing infinite storage capacity. Thus, if a work-piece routed to a particular cell finds that the cell is full, it refuses entry and it is routed back to the centralized storage area. If a work-piece routed by the conveyor is different from the required types to be sorted respectively, "a" and "b", it is rejected. We notice that once a work-piece is blocked from entry to a cell, the conveyor does not stop service; instead it proceeds to the other work-pieces waiting for transport. We also assume that within a cell no further blocking is caused once a work-piece is admitted. At the system level, we assume that the cells are functionally