

Compound Petri nets and alternatives aggregation Petri nets: Two formalisms for decision-making support

Advances in Mechanical Engineering
2016, Vol. 8(11) 1–12
© The Author(s) 2016
DOI: 10.1177/1687814016680516
aime.sagepub.com



Juan-Ignacio Latorre-Biel¹, Emilio Jiménez-Macías²,
Mercedes Pérez-de-la-Parte³, Juan Carlos Sáenz-Díez²,
Eduardo Martínez-Cámara³ and Julio Blanco-Fernández³

Abstract

Decision-making in technological systems, such as communication networks, manufacturing facilities and supply chains, constitutes a common requirement able to lead companies galore to success or failure. This article presents a decision-making methodology, where the feasible structural configurations to be analysed are chosen heuristically in the frame of a single optimization problem. For stating the optimization problem and solving it efficiently, appropriate formalisms would be used. Compound Petri nets, a particular kind of parametric Petri nets, and alternatives aggregation Petri nets, are two Petri net-based formalisms able to integrate in the same model different alternative structural configurations. Moreover, even having different characteristics that might make them useful for different applications, both formalisms present common features, such as including a set of exclusive entities and the possibility of developing compact Petri net models, by the removal of redundant information. This article is also focused on the transformation algorithm between compound Petri nets and alternatives aggregation Petri nets. This algorithm is devoted to transform a model described by one of the formalisms into an equivalent model, that is, with the same behaviour, represented using the other formalism. Finally, several application examples are given for illustrating the steps of the transformation algorithm.

Keywords

Alternatives aggregation Petri nets, compound Petri nets, parametric Petri nets, decision support systems, simulation, optimization

Date received: 6 November 2015; accepted: 1 November 2016

Academic Editor: ZhiWu Li

Introduction

Decision support systems

Decision support systems constitute tools of great help to the management of systems that show complex behaviour, arisen from the interrelations between the constituent subsystems. The main topic of this article contributes to the development process of efficient decision support systems under certain assumptions.

Since the 1960s, in the 20th century, the decision support systems have been an active research field. It has focused on the information management systems

¹Department of Mechanical, Energy and Materials Engineering, Public University of Navarre, Pamplona, Spain

²Department of Electrical Engineering, University of La Rioja, Logroño, Spain

³Department of Mechanical Engineering, University of La Rioja, Logroño, Spain

Corresponding author:

Emilio Jiménez-Macías, Department of Electrical Engineering, University of La Rioja, San Jose de Calasanz 31, Edificio Departamental, Logroño 26004, La Rioja, Spain.

Email: emilio.jimenez@unirioja.es



based on computers, which help in the decision-making process.

The decision support systems can be found in very diverse applications including the following: (a) medical diagnosis, (b) business management, (c) airports, as well as manufacturing and (d) supply chain planning. These systems are particularly useful in environments that change swiftly.

The type of system that will be considered in this article for decision-making is the discrete event system (DES).¹ The formal representation that has been used in this article for the DESs is the paradigm of the Petri nets (PNs).²⁻⁵

Our civilization has experienced a significant technological development and a generalization in the use of computers in the most diverse sectors. These facts have implied that a large number of technological systems can be modelled as DESs. Among them, it can be found manufacturing systems, supply chains, and communication networks (Recalde et al., 2003).⁶

Furthermore, the usual decisions involved in this type of systems, which have been analysed in the scientific literature include the following: (a) the design and (b) the control. In this context, control is understood as the process that allows a given system to show the desired behaviour.

For more details, the following references can be reviewed: Ramadge and Wonham,⁷ Lee and DiCesare,⁸ Moody and Antsaklis,⁹ Ghaffari et al.,¹⁰ Bai et al.¹¹ and Piera and Music.¹²

Constraint satisfaction and optimization problems

A problem of decision-making, expressed in a formal language, might lead to the statement of a constraint satisfaction problem. This situation will happen, in case that, for solving the problem; it is enough to find a solution satisfying a given set of equalities and inequalities. On the other hand, it might be required that in addition to finding a feasible solution, it can maximize or minimize one or several quality parameters of the feasible solutions. In this case, the problem is an optimization one.¹³

From now on, in this article, the optimization problem will be considered as the most general problem. It has to be considered that a solution of an optimization problem is one of a constraint satisfaction problem, as well. In particular, the constraint satisfaction problem is the one obtained by removing the objective or cost function from the former.

Simulation and choice of feasible solutions

A classic methodology for solving an optimization problem based on a DES consists of simulation. This

methodology requires the specification of certain conditions of the model of the system, with the purpose of setting values to the unknowns associated with its controllable parameters. In this article, the term 'undefined' will be applied to any characteristic of the PN model or the original system, where unknowns can be found. In fact, these unknowns are associated with the freedom degrees that require a decision-making process for defining or operating the system. According to this idea, an undefined structure of a system implies that the physical constitution of the system, not necessarily its behaviour, presents freedom degrees. An undefined parameter consists of a controllable parameter of the PN model prior to a decision that specifies a value for it.

A feasible solution of the problem should be chosen, and the evolution of the system under the desired constraints should be analysed. After performing a certain number of selections and their corresponding simulations, the solution of the problem that has led to a behaviour of better quality may be the final solution for the problem.¹⁴

Usually, the analysis of all the feasible solutions must be discarded by practical reasons. This fact can be explained because (a) the domain of the cost or objective function, also called solution space of the optimization problem, may be very large and (b) the computational cost of simulating a single solution may be also very high.

A significant limitation of simulation as optimization tool is the choice of the feasible solutions that will be considered for defining the simulation constraints.

If the selection of the feasible solutions has not been adequate and the best solutions are not chosen, the outcome of the simulation-based optimization process might not be a successful solution for the problem.

The process of searching for feasible solutions for defining the constraints of simulation can be guided, in an algorithmic procedure, by means of a probabilistic technique, such as a metaheuristic (Baruwa et al., 2015).¹⁵ The model of the system can be represented by a type of parametric PN called compound PN, where the controllable parameters of the system are represented in a symbolic way. These parameters should be specified, as a result of the searching algorithm, by real values corresponding to the chosen solution.

Several methods can be applied for guiding the search process, such as genetic algorithms, simulated annealing, ant colony and tabu search. These algorithms explore the solution space searching for solutions that might lead to optimal values of the cost function.¹⁶

The probabilistic nature of these methods allows escaping from local optima in their search for the global optimum. Nevertheless, metaheuristics do not guarantee that the optimum of the problem is going to be found in a given application of the methodology.

The problem of designing a new system

The DES, whose design or operation is the objective of the stated decision problem, might, in particular, present freedom degrees in its structure. This situation is very common in processes, where the system is under a process of design; hence, the lack of definition may cover many parts of the system itself. For example, there may exist a variety of alternative structural configurations for the system. Any of these structures can lead to very different models and, as a consequence, to systems of wide diversity.

This situation might arise in a design process, where different structural solutions can be taken into account. For example, it is possible to consider the design of a manufacturing system, where different suppliers may provide with different solutions for every subsystem: raw materials supply, manufacturing lines or cells, machining centres, assembly stations, quality control, reject line, storage and so on.

In this case, it is possible to build up diverse alternative solutions for the manufacturing system by means of a combination of the diverse subsystems provided by the different suppliers.

The design problem, sometimes, might produce a number of structural alternatives as a consequence of the combinatorial process arisen from the consideration of the different solutions for the subsystems.¹⁷

The combinatorial process for constructing the feasible solutions of the decision problem may lead to a very large solution space, a situation called combinatorial explosion. This fact, in its own, justifies the requirement of particular search strategies to find promising solutions among the set of feasible solutions, impossible to explore exhaustively.

On the other hand, the structural freedom degrees can arise in other situations than in design of DESs, for example, in a control process. An example of the former is the choice in real time of a manufacturing strategy. This choice may imply changes in material and information routes, in product lot and conveying sizes and so on.

Anyway, the PN models developed for processes, where there is a diversity of alternative structural configurations, may present the following: (a) diverse sizes for the incidence matrices and (b) different values for their elements, associated with the alternative structures.

Classic solving approach: divide and conquer

A common approach for coping with the optimization of this type of systems consists of (a) selecting some of the feasible structural configurations and (b) performing simulations or specific optimization processes for each case. Following this point of view, eventually, it is

possible to characterize every alternative structural configuration by a quality parameter resulting from the behaviour of the system.

In optimization problems, where there are not any undefined structural parameters, there is a single structural configuration for the system.¹⁸ This kind of problem is a classic one, where an optimal solution can be found by means of the optimization methodology described in the following. The stages of the solution process for the optimization problems are as follows: (a) choosing a subset of promising solutions from the solution space, (b) simulating their behaviour calculating a quality parameter, (c) comparing these parameters and (d) selecting the best solution.

The classic approach to solve a similar problem with alternative structural configurations consists of the following: (a) a first step of selecting the most promising structures and (b) launching an optimization process. This process, as described in the previous paragraphs, is performed by the association of the DES with every chosen alternative structure.

Proposed solving methodology: unity makes strength

Alternatively, this article proposes a solving strategy for optimization problems with alternative structural configurations. This methodology is based on the application of the classic methodology to a case, where there is an absence of structural parameters. Instead of stating so many optimization problems as alternative structures, a single optimization process will be stated for the whole decision-making process. The possibility for doing so is based on the use of the appropriate formalisms. These proposed formalisms are included in the model itself, the complete set of alternative structural configurations: (a) compound PNs and (b) alternatives aggregation Petri nets (AAPNs).

As a first advantage of this strategy, it can be stated that it reduces the limitations in the process of intuitive selection of the structural alternatives of the system.

Furthermore, the optimization in a single process prevents the dead ends associated with the optimization processes related to specific non-successful structures. Both the proposed methodology and the classic approach often require making different types of decisions. These decisions are related not only to the structure of the system but also to other controllable non-structural parameters, such as timing and initial markings.

The main drawback of this proposed strategy is the fact that the PN model uses to be larger and more complex than a single model constructed for a unique alternative structural configuration. However, the amount of information required to describe the single PN model may be smaller than the information required to describe the set of PN models associated with different

alternative structures. This reduction depends mainly in the similarity of the diverse alternative structures. The mentioned similarity is common in certain situations such as the design of DESs.^{19,20}

The comparison between the performance of both the proposed and the classic methodologies is favourable to the former in case there are similarities among the alternative structural configurations.

This article deals with two formalisms, based on the paradigm of the PNs. Both formalisms have been created for modelling, in a compact way, a DES with structural freedom degrees. This compactness of the model allows developing optimization procedures that, in certain cases, will be more efficient than the classic methodology.

Compound and AAPNs

The two formalisms that will be considered in this article are the compound PNs and the AAPNs. The transformation between both formalisms allows defining the equivalence conditions among models described with these different formalisms. It also allows comparing the performance of both formalisms applied to the same optimization problem. Two early versions of transformation algorithms between these two formalisms, forward and backward, were presented by Latorre-Biel and Jiménez-Macías;²¹ however, in this article, a new, improved and complete version of the transformation algorithm, from a compound PN to an AAPN, as well as the mathematical foundations of the complete algorithm is presented and detailed.

On the other hand, the differences among both formalisms, the compound PNs and the AAPNs, can be used in diverse applications or in different stages of the same application. In this last step, it is very convenient to resort to a transformation algorithm, such as the one presented in this article. This operation might profit from one of the formalisms in a certain stage of the resolution of the decision-making problem and from the other formalism in a subsequent stage.

The structure of this article is described in this paragraph. In section ‘Modelling of a DES with an undefined structure’, several previous concepts about modelling of systems with structural undefinition, including the compound PNs and the AAPNs, will be presented. Section ‘Transformation algorithm from a compound PN to an AAPN’ is focused on the transformation algorithm to obtain an equivalent model represented by means of an AAPN, from a given model, described using the formalism of the compound PNs. Some examples to illustrate the application of this technique are presented in section ‘Examples of application’, while section ‘Application of the models to a decision-making process’ summarizes the steps for the application of the proposed decision-making process.

Finally, section ‘Conclusion and future research’ deals with the conclusions and future research lines.

Modelling of a DES with an undefined structure

A compound PN is a formalism based on the paradigm of the PNs. It allows modelling a DES with an undefined structure by introducing parameters in its incidence matrices. Although an undefined PN can be defined as a PN with undefinition in any of its components (structure, marking, time, interpretation etc.), the article deals with two types of PNs with structural undefinition: compound PNs and AAPNs.

Definition 1. Compound PN. A compound PN R^c can be defined as a 7-tuple $R^c = \langle P, T, \text{Pre}, \text{Post}, \mathbf{m}_0, S_\alpha, S_{val\alpha} \rangle$, where

1. P and T are a set of places and transitions, respectively, such that $P \neq \emptyset$ and $T \neq \emptyset$.
2. Pre and Post are the pre-incidence and post-incidence matrices, respectively.
3. \mathbf{m}_0 is the initial marking.
4. $S_\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ is a set of controllable undefined parameters.
5. $S_{val\alpha}$ is the set of feasible combinations of values for the controllable undefined parameters. $S_{val\alpha} = \{cv_1, cv_2, \dots, cv_m\}$ verifies that the assignment $(\alpha_1, \alpha_2, \dots, \alpha_n) = cv_i$ should be performed as a consequence of a decision, which discards the other elements belonging to $S_{val\alpha}$ that have not been chosen.
6. $|S_{str\alpha}| > 1$, that is to say, in order the PN to be a compound PN and not only a parametric PN, it is necessary that it includes at least a controllable undefined structural parameter. □

Notice that the condition 5 of the *Definition 1* implies that the choice of a certain structure given by $cv_{strj} \in S_{valstr\alpha} = \{cv_{str1}, cv_{str2}, \dots, cv_{strm}\}$ is a consequence of a decision. Summarizing, it can be found that $S_{valstr\alpha}$ is a set of exclusive entities that reflects the nature, in turn exclusive, of the different structural alternatives for the DES with an undefined structure.

The compound PN allows to profit from the similarities or shared information among the alternative structural configurations of the DES. However, the existence of a number of undefined structural parameters, as well as of a large set of combinations of values for them, might lead the simulation of the model of the system to require a significant usage of computer resources.

In order to provide with another formalism that can be built up, without including parameters in the

incidence matrices, by means of different modelling procedures, the AAPNs have been created. The AAPNs present a conceptual proximity with the coloured PNs.

The first element of an AAPN bears the exclusive nature of the structural alternatives modelled by the net.

Definition 2. Choice variables. A set of choice variables is defined in the following way

$$S_A = \{a_1, a_2, \dots, a_s \mid \exists! a_i = 1, \text{ where } i \in N^*, \\ 1 \leq i \leq s \wedge a_j = 0 \forall j \neq i, 1 \leq j \leq s\}$$

Moreover, the assignment $a_i = 1$ is the result of a decision. \square

As a consequence, the role that $S_{valstr\alpha}$ plays in a compound PN, in the case of an AAPN is carried out by S_A .

Definition 3. AAPN. An AAPN, R^A , can be defined as the following 9-tuple

$$R^A = \langle P, T, \text{Pre}, \text{Post}, \mathbf{m}_0, S_\alpha, S_{val\alpha}, S_A, f_A \rangle$$

where

1. P and T are a set of places and transitions, respectively, which verify $P \neq \emptyset$ and $T \neq \emptyset$.
2. Pre and Post are the pre-incidence and post-incidence matrices, respectively.
3. \mathbf{m}_0 is the initial marking.
4. $S_\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ is a set of controllable undefined parameters. In an AAPN, it might happen that $S_{str\alpha} = \emptyset$.
5. $S_{val\alpha}$ is the set of feasible combinations of values for the controllable undefined parameters.
6. S_A is a set of choice variables, such that $|S_A| > 1$.
7. $f_A: T \rightarrow f(a_1, \dots, a_n)$ is a function that assigns to every transition t a Boolean function of choice variables. \square

In the previous definition, the role played by f_A guarantees that every choice variable can be associated with a set of transitions. This choice variable would be evaluated before the evolution of the system begins, once that a decision has allowed activating one of the choice variables.

It is interesting to consider that the set of undefined structural parameters of R^A is a subset of the undefined parameters: $S_{str\alpha} \subseteq S_\alpha$. If $S_{str\alpha} = \emptyset$, then R^A is called a simple AAPN, while in case that $S_{str\alpha} \neq \emptyset$, then R^A is a compound AAPN.

As a consequence of the previous considerations, it can be seen that the evolution rules for an AAPN change slightly from the dynamics of a generalized PN.

In particular, before the AAPN begins its evolution, a decision must be made concerning which choice variable should be activated. This decision assigns '0', meaning not chosen, to all the other choice variables.

Definition 4. Enabled transition of an AAPN. Let us consider an AAPN R^A associated with a set of choice variables $S_A = \{a_1, a_2, \dots, a_s\}$; let us consider the following decision

$$a_i = 1 \Rightarrow a_j = 0 \forall j \in N^* \text{ such that } 1 \leq j \leq n \wedge j \neq i$$

A certain transition $t_j \in T$ of an AAPN is said to be enabled if

$$m_i \geq \text{Pre}(p_i, t_j) \forall p_i \in {}^\circ t_j \wedge f_A(t_j) = 1$$

\square

In other words, transition t_j belonging to the PNs enabled when the number of tokens in place p_i is at least as large as the weight of the arc connecting p_i with t_j , for every place p_i which are input places to the transition t_j , and the function of choice variables associated with this transition leads to a true value if evaluated.

It should be noticed that it is not necessary to evaluate $f_A(t_j)$ in every step of the evolution of the AAPN. It is possible to evaluate every function of choice variables associated with the transitions before starting the simulation and to remove every transition associated with a function of choice variables with value False.

On the other hand, the transitions that are associated with a function of choice variables, whose value is True, should be preserved, but the function can be removed.

Transformation algorithm from a compound PN to an AAPN

Let us consider the following compound PN $R^c = \langle P, T, \text{Pre}, \text{Post}, \mathbf{m}_0, S_\alpha, S_{val\alpha} \rangle$:

Step 1. Define a set of choice variables from $S_{valstr\alpha}(R^c) = \{cv_{str1}, cv_{str2}, \dots, cv_{strm}\}$ in the form $S_A = \{a_1, a_2, \dots, a_m \mid \exists! a_i = 1, 1 \leq i \leq m, a_j = 0 \forall j \neq i, 1 \leq j \leq m\}$, where $|S_A| = |S_{valstr\alpha}(R^c)| = m$.

Step 2. Create a bijection among the elements of $S_{valstr\alpha}(R^c)$ and the choice variables of S_A .

Step 3. Replicate every transition t_i in the set $\{t_i^1, t_i^2, \dots, t_i^m\}$, where $m = |S_A|$ and

1. $\text{Pre}(p_j, t_i^q), \text{Post}(t_i^q, p_k) \subseteq (v_q^1, v_q^2, \dots, v_q^m)$ containing the values for every structural parameter associated with the combination of values for the undefined parameters cv_{strq} .

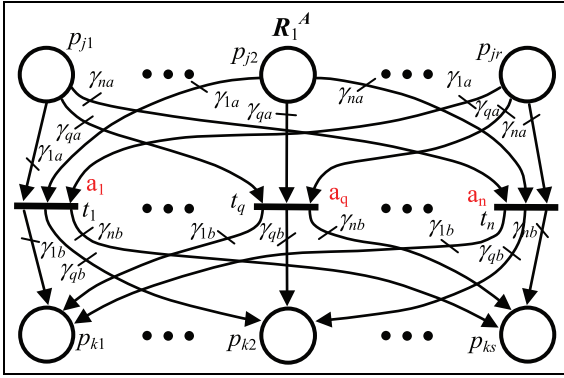


Figure 1. Subnet associated with the set of quasi-identical transitions t_1, t_2, \dots, t_n .

2. The choice variable a_q is associated with the transition t_i^q as a Boolean condition for controlling the enabling of the transition.

Step 4. Transform the AAPN resulting from the application of step 3 removing the isolated transitions and places.

Step 5. Application of a reduction rule to quasi-identical transitions, associated with different choice variables, thus obtaining an AAPN with a reduced set of transitions. \square

The step 5 of the previous algorithm requires to define the concept of quasi-identical transitions, see Figure 1, and to formulate the reduction rule.

Definition 5. Quasi-identical transition. Let $t_1, t_2, \dots, t_n \in T(R^A)$ be a set of transitions of an AAPN. A pair of transitions belonging to $T(R^A)$ is said to be quasi-identical if they verify the following properties:

1. $\forall t_i, t_j \in T(R^A) \exists a_i, a_j \in S_A$ such that a_i is associated with t_i , and a_j is associated with t_j , where $t_i \neq t_j$.
2. $\forall t_i, t_j \in T(R^A)$ if it is removed the association with a_i and $a_j \Rightarrow t_i$ and t_j are identical transitions.²² \square

Proposition. Reduction rule for quasi-identical transitions. Let R^A be an AAPN and S_A its associated set of choice variables.

Let $t_1, t_2, \dots, t_n \in T(R^A)$ be n quasi-identical transitions of the AAPN, such that $\forall t_i \in T(R^A)$, where $1 \leq i \leq n$, a_i is associated with t_i , with $a_i \in S_A$ (see Figure 1).

Let t be, without associated choice variables, an identical transition to t_1, t_2, \dots, t_n . If the function of choice variables $a_1 + a_2 + \dots + a_n$, where $+$ is the logic operator 'OR', is associated with t , then it is possible to state that t is equivalent to t_1, t_2, \dots, t_n .

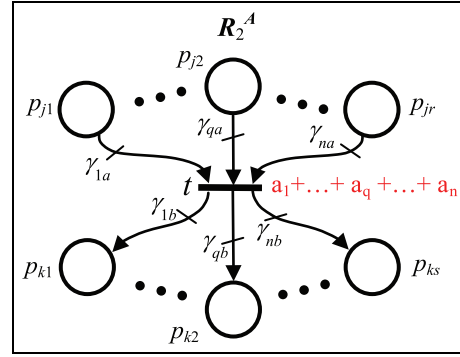


Figure 2. Subnet associated with the transition t .

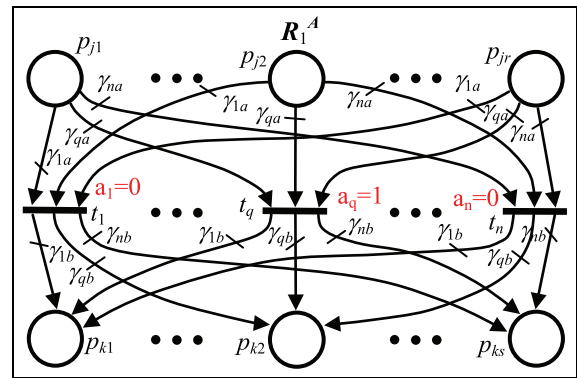


Figure 3. Subnet associated with the set of choice variables t_1, t_2, \dots, t_n after the activation of the choice variable a_q .

Proof. In order to analyse the behaviour of both equivalent subnets, one related to transition t , see Figure 2, and other associated with the set of quasi-identical transitions t_1, t_2, \dots, t_n , see Figure 1, it is necessary, first of all, to select arbitrarily one of the choice variables to be activated. As a result of this selection, the choice variable $a_q \in S_A$ will be activated, with $1 \leq q \leq n$.

By definition of choice variables, the rest of the choice variables $a_i \in S_A$, $i \neq q$, $1 \leq i \leq n$, will verify that $a_i = 0$.

This fact implies that in the evolution of the AAPNs, where the set of identical transitions t_1, t_2, \dots, t_n exists, only t_q can be enabled, hence, the rest of them can be removed (see Figure 3).

On the other hand, the function of choice variables $a_1 + a_2 + \dots + a_n$ associated with t will verify that $a_1 + a_2 + \dots + a_n = 0 + \dots + 0 + a_q + 0 + \dots + 0 = 1$ (see Figure 4).

Due to the fact that t and t_q are identical transitions, and they are both associated with an activated choice variable or function of choice variables, their behaviour is the same. This reasoning can be repeated for any of the other transitions of the set t_1, t_2, \dots, t_n with the same results. As a consequence, t_1, t_2, \dots, t_n and t are equivalent.

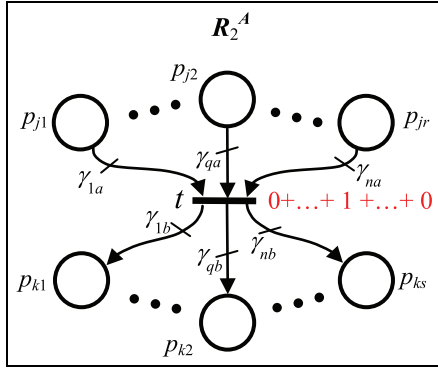


Figure 4. Subnet associated with the transition t after the activation of the choice variable a_q .

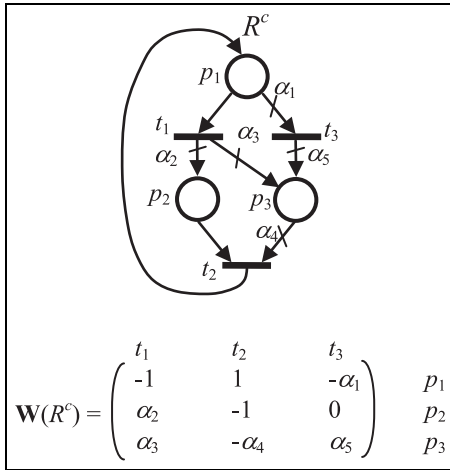


Figure 5. Compound Petri net.

Examples of application

Example 1

Let us consider the compound PN depicted in Figure 5.

In this first example, a compound PN will be transformed into an equivalent AAPN by means of the application of the steps of the algorithm described in the previous section.

The set of undefined parameters of the original compound PN contains only five structural parameters $S_\alpha(R^c) = S_{str\alpha}(R^c) = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}$. On the other hand, the set of feasible combinations of values for the undefined structural parameters has three elements: $S_{valstr\alpha}(R^c) = \{(0,2,0,0,0), (0,1,1,1,0), (1,2,0,1,2)\}$:

$$\text{Step 1. } S_{valstr\alpha}(R^c) = \{(0,2,0,0,0), (0,1,1,1,0), (1,2,0,1,2)\}, |S_{valstr\alpha}(R^c)| = 3 = |S_A|.$$

Hence, it is possible to create the following set of choice variables

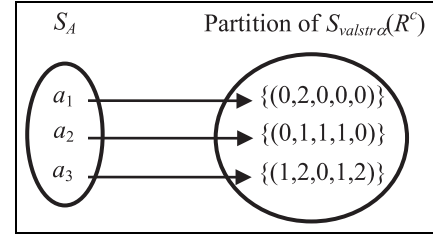


Figure 6. Bijection between S_A and $S_{valstr\alpha}(R^c)$.

$$\mathbf{W}(R^c) = \begin{pmatrix} t_1 & t_2 & t_3 \\ -1 & 1 & -\alpha_1 \\ \alpha_2 & -1 & 0 \\ \alpha_3 & -\alpha_4 & \alpha_5 \end{pmatrix} \begin{matrix} p_1 \\ p_2 \\ p_3 \end{matrix} = \begin{pmatrix} t_1 & t_2 & t_3 \\ -\beta_6 & \beta_7 & -\alpha_1 \\ \alpha_2 & -\beta_8 & \beta_9 \\ \alpha_3 & -\alpha_4 & \alpha_5 \end{pmatrix} \begin{matrix} p_1 \\ p_2 \\ p_3 \end{matrix}$$

Where α_i is an undefined structural parameter.
 β_j is a defined structural parameter, whose value is unique and known.

Figure 7. Naming of the structural parameters of R^c .

$$S_A = \{a_1, a_2, a_3 | \exists! a_i = 1, 1 \leq i \leq 3, a_j = 0 \forall j \neq i, 1 \leq j \leq 3\}$$

Step 2. Create a bijection between the elements of $S_{valstr\alpha}(R^c)$ and S_A (see Figure 6).

Step 3. Replicate every transition t_i in the set $\{t_i^1, t_i^2, \dots, t_i^m\}$, where $3 = |S_A|$ and $1 \leq i \leq 3$.

1. $\text{Pre}(p_j, t_i^q), \text{Post}(t_i^q, p_k) \subseteq (v_q^1, v_q^2, \dots, v_q^m)$ containing the values for every structural parameters associated with the combination of values for the undefined parameters cv_{strq} .

In order to calculate the weight of the arcs of every replicated transitions, the structural parameters associated with every $cv_{strq} \in S_{valstr\alpha}(R^c) = \{cv_{str1}, cv_{str2}, \dots, cv_{strm}\}$ with $1 \leq q \leq 3$ will be considered. On the other hand

$$S_{str\alpha} = \{(a_1, a_2, a_3, a_4, a_5)\}$$

$$cv_{str1} = (0, 2, 0, 0, 0),$$

$$cv_{str2} = (0, 1, 1, 1, 0), \quad cv_{str3} = (1, 2, 0, 1, 2)$$

The values of the defined structural parameters defined of R^c are given by the following expression (see Figure 7)

$$S_{str\beta} = \{(\beta_6, \beta_7, \beta_8, \beta_9)\},$$

where $(\beta_6, \beta_7, \beta_8, \beta_9) = (1, 1, 1, 0)$

Transition t_1 is replicated in $\{t_1^1, t_1^2, t_1^3\}$. In this case, $i = 1$

$$\begin{aligned} q = 1 &\Rightarrow (v_q^1, v_q^2, \dots, v_q^m) \\ &= (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \beta_6, \beta_7, \beta_8, \beta_9) \\ &= (0, 2, 0, 0, 0, 1, 1, 1, 0) \end{aligned}$$

$$j = 1 \Rightarrow \text{Pre}(p_j, t_i^q) = \text{Pre}(p_1, t_1^1) = \beta_6 = 1$$

$$k = 2 \Rightarrow \text{Post}(t_i^q, p_k) = \text{Post}(t_1^1, p_2) = \alpha_2 = 2$$

$$k = 3 \Rightarrow \text{Post}(t_i^q, p_k) = \text{Post}(t_1^1, p_3) = \alpha_3 = 0$$

$$\begin{aligned} q = 2 &\Rightarrow (v_q^1, v_q^2, \dots, v_q^m) \\ &= (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \beta_6, \beta_7, \beta_8, \beta_9) \\ &= (0, 1, 1, 1, 0, 1, 1, 1, 0) \end{aligned}$$

$$j = 1 \Rightarrow \text{Pre}(p_j, t_i^q) = \text{Pre}(p_1, t_1^2) = \beta_6 = 1$$

$$k = 2 \Rightarrow \text{Post}(t_i^q, p_k) = \text{Post}(t_1^2, p_2) = \alpha_2 = 1$$

$$k = 3 \Rightarrow \text{Post}(t_i^q, p_k) = \text{Post}(t_1^2, p_3) = \alpha_3 = 1$$

$$\begin{aligned} q = 3 &\Rightarrow (v_q^1, v_q^2, \dots, v_q^m) \\ &= (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \beta_6, \beta_7, \beta_8, \beta_9) \\ &= (1, 2, 0, 1, 2, 1, 1, 1, 0) \end{aligned}$$

$$j = 1 \Rightarrow \text{Pre}(p_j, t_i^q) = \text{Pre}(p_1, t_1^3) = \beta_6 = 1$$

$$k = 2 \Rightarrow \text{Post}(t_i^q, p_k) = \text{Post}(t_1^3, p_2) = \alpha_2 = 2$$

$$k = 3 \Rightarrow \text{Post}(t_i^q, p_k) = \text{Post}(t_1^3, p_3) = \alpha_3 = 0$$

The process is repeated with the replication of transition t_2 in $\{t_2^1, t_2^2, t_2^3\}$ and t_3 in $\{t_3^1, t_3^2, t_3^3\}$:

- As a second stage in the step 3 of the transformation algorithm, the choice variable a_q is associated with the transition t_i^q as a Boolean condition that controls the enabling of the transition $1 \leq i \leq 3$.

The result of step 3 of the algorithm, consisting of the replication of every transition, can be seen in Figure 8.

It can be seen in the resulting AAPNs that it lacks undefined structural parameters in contrast with the original compound PN.

Furthermore, the association of the choice variables with the replicated transitions models the exclusiveness of the different alternative structural configurations. This exclusiveness requires making a decision before the evolution of the PN begins:

Step 4. Removal of the isolated places and transitions.

The isolated places and transitions lack of arc connecting them to other nodes of the net. For this reason, they do not intervene in the flow of tokens that describes the

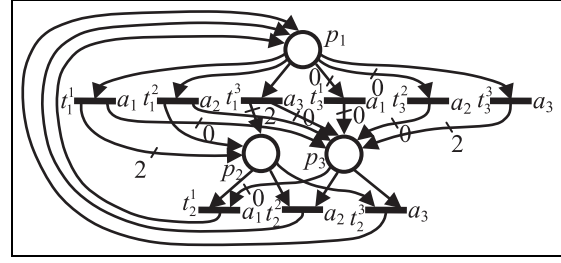


Figure 8. Graphical representation of R^c after replicating the transitions.

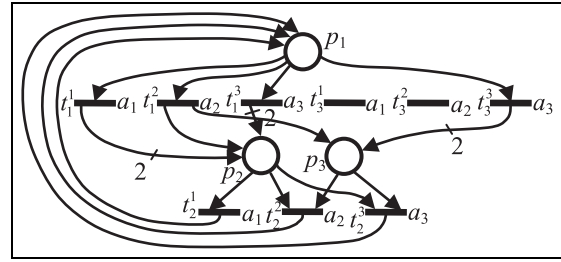


Figure 9. Alternatives aggregation Petri nets after removing the zero-weighted arcs.

evolution of the system. As a consequence, the isolated places and transitions can be removed from the AAPN without altering its behaviour.

The first stage of step 4 consists of identifying and removing from the AAPN all the arcs with weight zero. The result can be seen in Figure 9.

As a next step of step 4, the isolated places and transitions should be identified and removed. In this example, there are two isolated transitions: t_3^1 and t_3^2 . In order to conclude step 4, the isolated transitions should be removed. The result can be seen in Figure 10.

Step 5. Apply the reduction rule to the quasi-identical transitions, obtaining an AAPN with a reduced set of transitions.

In the reduction rule for quasi-identical transitions, it can be seen that a set of quasi-identical transitions can be reduced to a single transition associated with a certain function of choice variables.

In order to find quasi-identical transitions, it is necessary to search for identical transitions in the generalized PN obtained by removing the choice variables of the AAPN.

In the identical transitions, the input and output arcs should connect the transitions with the same places and should be weighted with the same values. On a further step, it should be checked whether the associated choice variables verify the condition of the reduction rule.

In the present example, two pairs of quasi-identical transitions have been found: t_1^1 and t_1^3 on one side and

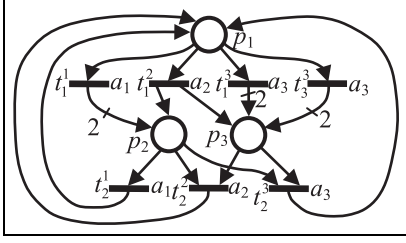


Figure 10. Alternatives aggregation Petri net after the application of step 4.

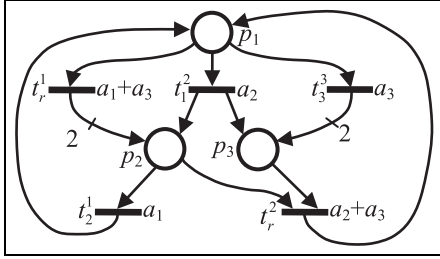


Figure 11. Alternatives aggregation Petri net, equivalent to the compound Petri net R^c .

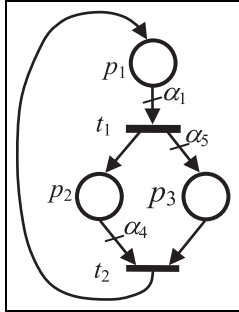


Figure 12. Compound PN.

t_2^2 and t_2^3 on the other. The application of the reduction rule allows the substitution of the first pair for the transition t_r^1 , which is associated with the function of choice variables $a_1 + a_3$. Moreover, the second pair of quasi-identical transitions can be replaced by t_r^2 , associated with the function $a_2 + a_3$. The result of this last step of the algorithm is depicted in Figure 11.

Example 2

In the following, a compound PN in both representations can be seen: a graphical one in Figure 12 and another matrix-based one based on the incidence matrix in Figure 13.

The set of structural parameters of the compound PN R^c is as follows

$$\mathbf{W}(R^c) = \begin{pmatrix} t_1 & t_2 \\ -\alpha_1 & \beta_2 \\ \beta_3 & -\alpha_4 \\ \alpha_5 & \beta_6 \end{pmatrix} \begin{matrix} p_1 \\ p_2 \\ p_3 \end{matrix}$$

$$= \begin{pmatrix} t_1 & t_2 \\ -\alpha_1 & 1 \\ 1 & -\alpha_4 \\ \alpha_5 & 1 \end{pmatrix} \begin{matrix} p_1 \\ p_2 \\ p_3 \end{matrix}$$

Figure 13. Matrix-based representation of the compound Petri net.

$$\begin{aligned} S_{str\gamma}(R^c) &= \{\alpha_1, \beta_2, \beta_3, \alpha_4, \alpha_5, \beta_6\} \\ &= S_{str\alpha}(R^c) \cup S_{str\beta}(R^c), \text{ where} \\ &= S_{str\alpha}(R^c) = \{\alpha_1, \alpha_4, \alpha_5\} \text{ and } S_{str\beta}(R^c) \\ &= \{\beta_2, \beta_3, \beta_6\} \end{aligned}$$

On the other hand, the set of feasible values for the structural parameters of R^c can be written as follows

$$\begin{aligned} S_{valstr\beta}(R^c) &= \{(1, 1, 1)\} \\ S_{valstr\alpha}(R^c) &= \{(1, 0, 1), (0, 1, 0), (2, 0, 1), (0, 1, 2)\} \end{aligned}$$

Finally, it is possible to determine the set of feasible values for every undefined structural parameter of R^c

$$\begin{aligned} S_{val\alpha_1}(R^c) &= \{0, 1, 2\} \\ S_{val\alpha_4}(R^c) &= \{0, 1\} \\ S_{val\alpha_5}(R^c) &= \{0, 1, 2\} \end{aligned}$$

The first partition of $S_{valstr\alpha}(R^c)$ has order 2 and can be represented as follows

$$\begin{aligned} \prod_1(S_{valstr\alpha}(R^c)) &= \{S_{1valstr\alpha}(R^c), S_{2valstr\alpha}(R^c)\} \\ S_{valstr\alpha}(R^c) &= S_{1valstr\alpha}(R^c) \cup S_{2valstr\alpha}(R^c) \\ S_{valstr\alpha}(R^c) &= \{(1, 0, 1), (0, 1, 0), (2, 0, 1), (0, 1, 2)\} \\ S_{1valstr\alpha}(R^c) &= \{(1, 0, 1), (0, 1, 0)\} \\ S_{2valstr\alpha}(R^c) &= \{(2, 0, 1), (0, 1, 2)\} \end{aligned}$$

In order to know the number of undefined structural parameters associated with every subset of the partition, it is necessary to analyse every parameter of $S_{valstr\alpha}(R^c)$.

On one hand, the first subset of the partition will be considered

$$\begin{aligned} S_{1val\alpha_1}(R^c) &= \{0, 1\} \\ S_{1val\alpha_4}(R^c) &= \{0, 1\} \\ S_{1val\alpha_5}(R^c) &= \{0, 1\} \end{aligned}$$

As a consequence, there will be three undefined structural parameters associated with this subset of the partition, since any of them can take values from a set with more than one element

$$S_{1str\alpha}(R^c) = \{\alpha_1^1, \alpha_4^1, \alpha_5^1\}$$

On the other hand, the second subset of the partition will lead to the following

$$S_{2val\alpha_1}(R^c) = \{0, 2\}$$

$$S_{2val\alpha_4}(R^c) = \{0, 1\}$$

$$S_{2val\alpha_5}(R^c) = \{1, 2\}$$

In this case, there will be another new three undefined structural parameters associated with this subset of the partition, since any of them can take values from a set with more than one element

$$S_{2val\alpha}(R^c) = \{\alpha_1^2, \alpha_4^2, \alpha_5^2\}$$

As a result, it is possible to see how this partition of $S_{valstr\alpha}(R^c)$, $\Pi_1(S_{valstr\alpha}(R^c))$, from a compound PN with three undefined structural parameters leads to a representation with six undefined structural parameters. This representation can be a set of compound alternative PN or an AAPN. The AAPN that results from the replication of the transitions of the compound PN R^c according to this partition is represented in Figure 14(a) and its incidence matrix is shown in Figure 14(b).

As a conclusion, it is possible to state that despite the fact that the original compound PN R^c has only three undefined structural parameters $S_{str\alpha}(R^c) = \{\alpha_1, \alpha_4, \alpha_5\}$, the resulting AAPN obtained by a replication of the transitions based on this first partition of $S_{valstr\alpha}(R^c)$ has six undefined structural parameters

$$S_{str\alpha}(R^A) = S_{1str\alpha}(R^c) \cup S_{2str\alpha}(R^c) = \{\alpha_1^1, \alpha_4^1, \alpha_5^1, \alpha_1^2, \alpha_4^2, \alpha_5^2\}$$

Example 3

The second partition of $S_{valstr\alpha}(R^c)$ has order 2 and can be represented as follows

$$\prod_2(S_{valstr\alpha}(R^c)) = \{S_{1valstr\alpha}(R^c), S_{2valstr\alpha}(R^c)\}$$

$$S_{valstr\alpha}(R^c) = S_{1valstr\alpha}(R^c) \cup S_{2valstr\alpha}(R^c)$$

$$S_{valstr\alpha}(R^c) = \{(1, 0, 1), (0, 1, 0), (2, 0, 1), (0, 1, 2)\}$$

$$S_{1valstr\alpha}(R^c) = \{(1, 0, 1), (2, 0, 1)\}$$

$$S_{2valstr\alpha}(R^c) = \{(0, 1, 0), (0, 1, 2)\}$$

In order to know the number of undefined structural parameters associated with every subset of the partition, it is necessary to analyse every parameter of $S_{valstr\alpha}(R^c)$.

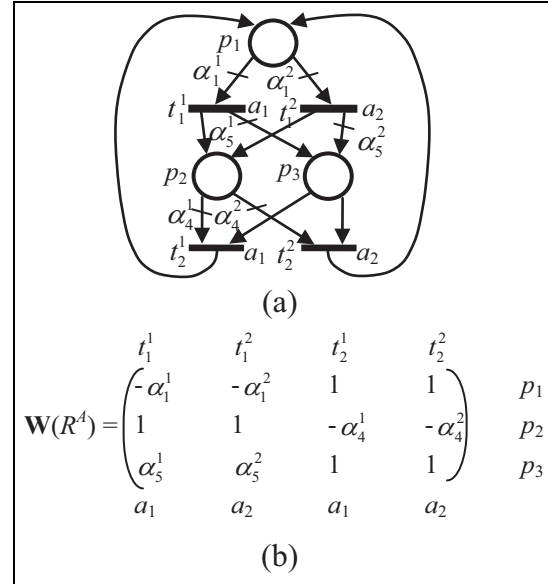


Figure 14. Graphical representation of an alternatives aggregation Petri net: (a) obtained from first partition of $S_{valstr\alpha}(R^c)$ and (b) matrix-based representation obtained from a first partition of $S_{valstr\alpha}(R^c)$.

On one hand, the first subset of the partition will be considered

$$S_{1val\alpha_1}(R^c) = \{1, 2\}$$

$S_{1val\alpha_4}(R^c) = \{0\} \Rightarrow$ In this subset of the second partition, α_4 is no longer an undefined structural parameter but a defined one: β_4^1 .

$S_{1val\alpha_5}(R^c) = \{1\} \Rightarrow$ In this subset of the second partition, α_5 is no longer an undefined structural parameter but a defined one: β_5^1 .

As a consequence, there will be only one undefined structural parameter associated with this subset of the partition, since only α_1^1 can take values from a set with more than one element

$$S_{1val\alpha_1}(R^c) = \{\alpha_1^1\}$$

On the other hand, the second subset of the partition will lead to the following:

$S_{2val\alpha_1}(R^c) = \{0\} \Rightarrow$ In this subset of the second partition, α_1 is no longer an undefined structural parameter but a defined one: β_1^1 .

$S_{2val\alpha_4}(R^c) = \{1\} \Rightarrow$ In this subset of the second partition, α_4 is no longer an undefined structural parameter but a defined one: β_4^1

$$S_{2val\alpha_5}(R^c) = \{0, 2\}$$

This case will provide with another single undefined structural parameter associated with the corresponding subset of this second partition, α_5^2 , since it is the only one that can take values from a set with more than one element

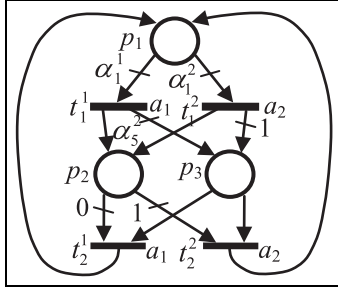


Figure 15. Graphical representation of an alternatives aggregation Petri net obtained from a second partition of $S_{valstr\alpha}(R^c)$.

$$W(R^A) = \begin{pmatrix} t_1^1 & t_1^2 & t_2^1 & t_2^2 \\ -\alpha_1^1 & -\alpha_1^2 & 1 & 1 \\ 1 & 1 & 0 & -1 \\ 1 & \alpha_5^2 & 1 & 1 \\ a_1 & a_2 & a_1 & a_2 \end{pmatrix} \begin{matrix} p_1 \\ p_2 \\ p_3 \end{matrix}$$

Figure 16. Matrix-based representation of an alternatives aggregation Petri net obtained from a second partition of $S_{valstr\alpha}(R^c)$.

$$S_{2val\alpha}(R^c) = \{\alpha_5^2\}$$

As a result, it is possible to see how this second partition of $S_{valstr\alpha}(R^c)$, $\Pi_2(S_{valstr\alpha}(R^c))$, from a compound PN with three undefined structural parameters leads to a representation with only two undefined structural parameters. This representation can be a set of compound alternative PN or an AAPN. The AAPN that results from the replication of the transitions of the compound PN R^c according to this partition is represented in Figures 15 and 16.

As a conclusion, it is possible to state that despite the fact that the original compound PN R^c has only three undefined structural parameters $S_{str\alpha}(R^c) = \{\alpha_1, \alpha_4, \alpha_5\}$, the resulting AAPN obtained by a replication of the transitions based on this second partition of $S_{valstr\alpha}(R^c)$ has only two undefined structural parameters

$$S_{str\alpha}(R^A) = S_{1str\alpha}(R^c) \cup S_{2str\alpha}(R^c) = \{\alpha_1^1, \alpha_5^2\}$$

It is interesting to notice that it depends on the parameters of the transformation algorithm (in this case the chosen partition) that the size of the resulting model is more or less compact.

Application of the models to a decision-making process

The process of decision-making associated with the previous examples might be carried out in the following way:

Step 1. Define the optimization problem to be solved:

Step 1.1. Determine the decision variables, also called undefined parameters of the PN model.

Step 1.2. Determine a quantitative method for assessing the quality of a given solution. For example, define a cost function.

Step 1.3. Determine the additional constraints of the decision problem. For instance, the time period of the PN and the model of a dynamic DES that covers the statement of the problem.

Step 2. Define the search method for feasible solutions in the solution space. For example, a meta-heuristic, such as genetic algorithms, tabu search, swarm particle, ant colony or simulated annealing.

Step 3. Launch an instance of the optimization problem:

Step 3.1. Select one feasible solution using the search algorithm.

Step 3.2. Simulate the behaviour of the PN model of the system under the chosen solution and the constraints of the problem.

Step 3.3. Calculate the quality of the simulated solution.

Step 3.4. If it is the first solution analysed, store it, otherwise, compare the quality of the new solution with the one of the stored solution and store the best of them.

Step 3.5. Evaluate the stopping criterion for deciding if a new solution should be evaluated (step 3.1) or the process is terminated. The solution of the optimization problem given by this technique is the best solution found by the algorithm.

This algorithm does not guarantee that the solution found is the optimum of the stated optimization problem. However, it is likely to be close to it; for this reason, it can be called quasi-optimal solution. The quality of this quasi-optimal solution will depend on the tuning of the search algorithm, as well as on the computational resources and time devoted to solve the problem.

Conclusion and future research

In this article, it has been shown how it is possible to relate two formalisms that have been created for representing a DES with an undefined structure, by means of a transformation algorithm. Both formalisms, based on the paradigm of the PN, the compound PNs and the AAPNs share the possibility of removing redundant information in models built up from alternative PNs.

Nevertheless, the compactness of the model in the case of the AAPNs is achieved by means of sets of structural parameters and the feasible combination for their values. On the other hand, the capacity for condensing information, in the case of the AAPNs, is based

on the existence of shared subnets and the possibility of controlling the token flow in the evolution of the PN by means of the choice variables.

Both formalisms, compound PNs and AAPNs, have led to promising results while implementing optimization algorithms with guided search by means of metaheuristics based on genetic algorithms.

However, there is a long way ahead in applying the methodology described in this article, in a variety of real cases, which will allow acquiring knowledge on the practical advantages and drawbacks of any of the presented formalisms.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship and/or publication of this article.

References

1. Cassandras Christos G and Lafortune S. *Introduction to discrete event systems*. 2nd ed. New York: Springer, 2008.
2. Peterson JL. *Petri net theory and the modeling of systems*. Englewood Cliffs, NJ: Prentice Hall, 1981.
3. Silva M. Introducing Petri nets. In: Di Cesare F (ed.) *Practice of Petri nets in manufacturing*. London: Chapman & Hall, 1993, pp.1–62.
4. Balbo G and Silva M (eds). *Performance models for discrete event systems with synchronizations: formalisms and analysis techniques*. Zaragoza: Editorial Kronos, 1998.
5. David R and Alla H. *Discrete, continuous and hybrid Petri nets*. Berlin: Springer, 2005.
6. Recalde L, Silva Suarez M, Ezpeleta J, et al. Petri nets and manufacturing systems: an examples-driven tour. *Lect Notes Comput Sc* 2003; 2003: 742–788.
7. Ramadge PJG and Wonham WM. The control of discrete event systems. *P IEEE* 1989; 77: 81–97.
8. Lee DY and DiCesare F. FMS scheduling using Petri nets and heuristic search. *IEEE T Robot Autom* 1992; 2: 1057–1062.
9. Moody JO and Antsaklis PJ. *Supervisory control of discrete event systems using Petri nets*. Norwell, MA: Kluwer, 1998.
10. Ghaffari A, Rezg N and Xie X. Design of a live and maximally permissive Petri net controller using the theory of regions. *IEEE T Robot Autom* 2003; 19: 137–142.
11. Bai Q, Ren F, Zhang M, et al. Using colored petri nets to predict future states in agent-based scheduling and planning systems. *J Multiagent Grid Syst: Adv Agent Mediat Autom Negot* 2010; 6: 527–542.
12. Piera MA and Music G. Coloured Petri net scheduling models: timed state space exploration shortages. *Math Comput Simulat* 2011; 82: 428–441.
13. Mujica Mota M and Piera MA. An improved time line search algorithm for manufacturing decision-making. *Int J Prod Res* 2014; 52: 1116–1132.
14. Juan AA, Faulin J, Grasman SE, et al. A review of simheuristics: extending metaheuristics to deal with stochastic combinatorial optimization problems. *Oper Res Perspect* 2015; 2: 62–72.
15. Baruwa OT, Piera MA and Guasch A. Deadlock-free scheduling method for flexible manufacturing systems based on timed colored petri nets and anytime heuristic search. *IEEE Trans Syst Man Cybern: Syst* 2015; 45(5): 831–846.
16. Latorre-Biel JI, Jiménez-Macías E and Pérez-Parte M. Sequence of decisions on discrete event systems modeled by Petri nets with structural alternative configurations. *J Comput Sci* 2014; 5: 387–394.
17. Latorre-Biel JI, Jiménez-Macías E and Pérez-Parte M. The alternatives aggregation Petri nets as a formalism to design discrete event systems. *Int J Simulat Process Model* 2010; 6: 152–164.
18. Zimmermann A, Rodriguez D and Silva M. A two phase optimization method for Petri net models of manufacturing systems. *J Intell Manuf* 2001; 12: 421–432.
19. Latorre-Biel JI, Jiménez-Macías E, Blanco-Fernández J, et al. Optimal design of an olive oil mill by means of the simulation of a Petri net model. *Int J Food Eng* 2014; 10: 573–582.
20. Latorre-Biel JI, Jiménez-Macías E, Pérez-de-la-Parte M, et al. Control of discrete event systems by means of discrete optimization and disjunctive colored PNs: application to manufacturing facilities. *Abstr Appl Anal* 2014; 2014: 821707.
21. Latorre-Biel JI and Jiménez-Macías E. Transformation algorithm from an alternatives aggregation Petri net to a compound Petri net. Two representations of an undefined Petri net with a non-empty set of exclusive entities. In: *Proceedings of the 24th European modeling and simulation symposium (EMSS 12)*, Vienna, 19–21 September 2012, pp.465–470. DIME/DIPTTEM, Università di Genova.
22. Berthelot G. Transformations and decompositions of nets. In: Brauer W, Reisig W and Rozenberg G (eds) *Petri nets: central models and their properties – advances in Petri nets*, vol. 254 (Lecture notes in computer science). Berlin: Springer, 1987, pp.359–376.