Int. J. Production Economics 143 (2013) 3–12

Contents lists available at ScienceDirect

Int. J. Production Economics

journal homepage: www.elsevier.com/locate/ijpe

Optimization of preventive maintenance through a combined maintenance-production simulation model

O. Roux [a,*], D. Duvivier [b,c], G. Quesnel [d], E. Ramat [b,c,d]

[a] Louvain School of Management and Catholic University of Mons, Chaussée de Binche, 151, B-7000 Mons, Belgium
[b] Univ Lille Nord de France, F-59000 Lille, France
[c] Université du Littoral Côte d'Opale, LISIC, BP 719, F-62228 Calais Cedex, France
[d] INRA, UR875 Biométrie et Intelligence Artificielle, F-31326 Castanet-Tolosan, France

ARTICLE INFO

Article history:
Received 8 January 2010
Accepted 4 November 2010
Available online 19 November 2010

Keywords:
Metaheuristics
Simulation
Decision-making
Multimodeling
Petri-nets
Maintenance

ABSTRACT

Maintenance problems are crucial aspect of nowadays industrial problems. However, the quest of the efficient periodicity of maintenance for all components of a system is far from an easy task to accomplish when considering all the antagonistic criteria of the maintenance and production views of a production system. Thus, the objective is to simultaneously ensure a low frequency of failures by an efficient periodic preventive maintenance and minimize the unavailability of the system due to preventive maintenance. This implies a minimum impact on the production. In this paper, several tools are combined to collaborate in order to optimize multi-component preventive maintenance problems. The structure of the maintenance-production system is modeled thanks to a framework inspired by our previous research projects. The dynamic aspects are modeled by a combination of timed petri-nets and PDEVS models and implemented in our VLE simulator. The parameters of the resulting simulation model are optimized via a Nelder–Mead (Simplex) Method.

1. Introduction

The present economical context requires from companies that they practice an optimal exploitation of their production tools. In this purpose, every decision maker is asked to assure a maximum availability of these production tools at minimal cost (Percy and Kobbacy, 2000). The optimization consists in determining the best "parameters combination" which provides the best values of the technical and economical criteria (see for instance Rezg et al., 2005; Boschian et al., 2009). However, in most cases, it appears to be very difficult to use analytical approaches without formulating restrictive hypotheses. In order to evaluate these performance criteria, simulation is the best adapted solution. In this paper, we suggest an approach integrating optimization and simulation. This approach consists in generating more and more efficient solutions with an optimization tool and to evaluate them via a simulation model until a halt criterion is satisfied. This approach has already been studied in the literature (see for instance Boschian et al., 2009; Riane et al., 2009). This integration is illustrated in Fig. 1. In the following sections, according to Talbi (2002), this assembly of different units, at different levels of combinations is called a "hybrid model".

Our work aims to provide a framework to facilitate the optimization of production and maintenance through simulation. This paper focuses on the simulation aspect. We want to develop a generic modeling tool for simulation, easy to understand by decision makers. The objective is to facilitate the creation of simulation models by the use of constructs (elementary components).

The remainder of this paper is organized as follows. The second section presents the maintenance problem; the third section introduces the simulation paradigms, formalisms and tools that constitute the bases of our framework; the fourth section depicts our modeling component; the fifth section describes an application of our optimization–simulation hybrid model. Finally several conclusions and perspectives are given.

2. Maintenance strategies

A maintenance strategy is defined as a decision rule which establishes the sequel of maintenance actions. Each maintenance action allows one to maintain or restore the system in a specified state by using the appropriate resources. Cost and duration are incurred to execute each maintenance action. Many papers dealing with preventive maintenance and replacement strategies have been published in the last two decades (Wang, 2002; Roux et al., 2008). We consider for this paper one basic replacement policy (Bloc Replacement Policy BRP). Barlow and Proshan (1976) consider also

* Corresponding author.
E-mail addresses: olivier.roux@fucam.ac.be (O. Roux), david.duvivier@lisic.univ-littoral.fr (D. Duvivier), gauthier.quesnel@toulouse.inra.fr (G. Quesnel), ramat@lisic.univ-littoral.fr (E. Ramat).

0925-5273/$ - see front matter
doi:10.1016/j.ijpe.2010.11.004

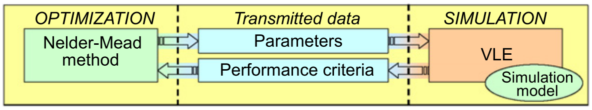

Fig. 1. Optimization and simulation integration.

the BRP where the replacements are undertaken at KT with $K=1,2,3,\ldots$ and T a fixed time, or at failure (see Fig. 2). Only new items are used to perform replacement. C_p and C_c are respectively the preventive and corrective replacement costs. Similarly T_p is defined as the duration for preventive maintenance action, and T_c as the duration for corrective maintenance operation. This maintenance strategy is also used in the simulation model described in the following sections.

In the result section, the availability is considered as the criterion to maximize. Effectively, since we are simultaneously considering production and maintenance in a context where production costs are higher than maintenance costs, the availability is a more adequate criterion than the maintenance costs. However, our model can be used to optimize the maintenance costs if needed.

3. Simulation

This section presents the VLE simulator and the underlying paradigm and formalism. This simulator relies on strong concepts and intrinsically provides multimodeling capabilities. This perfectly matches the objective of the simulation and modeling tool that we are currently implementing. This is also largely facilitated by the available extensions such as Petri-nets (Peterson, 1977).

3.1. DEVS, VLE and Petri-nets

Nowadays, it is recognized that multimodeling is a powerful concept for the modeling and simulation of large complex systems. At the end of 1980s, Fishwick and Zeigler (1992) introduced the multimodeling basis concepts. One can define multimodels as large models which are composed of different types of models (i.e. different paradigms or formalisms), (Fishwick, 1995). Concepts like refinement and hierarchical composition are basis of multimodeling. The first describes the decomposition of one model into several other ones in order to refine the behavior of the composed model. The last defines the opposite process: it is called models aggregation. In this context, a major issue is how to deal with the coupling of heterogeneous models. Several works dealing with the coupling of heterogeneous models have already been published. For a review of concepts and techniques, see the book of Zeigler et al. (2000). With DEVS, discrete event system specification, Zeigler (1976) has provided formal basis for the construction of coupled model in a network or graph manner. In this section, we focus on the discrete event system specification (DEVS) formalisms and their associated extensions, in particular Petri-nets.

3.1.1. Discrete event simulation

Our works take place in the Modeling and Simulation (M&S) theory defined by Zeigler (1976). M&S theory tends to be as general as possible. It addresses major issues of computer sciences. From artificial intelligence to model design and distributed simulations, M&S theory aims to develop a common framework (formal and operational) for the specification of dynamical systems. Many theoretical basis and formal extensions to DEVS were carried out, therefore, we advise the second edition of Zeigler et al.'s (2000) book to have an overall picture of these works. DEVS defines an atomic model as a set of input and output ports and a set of state transition functions: $M=\langle X,Y,S,\delta_{int},\delta_{ext},\lambda,ta\rangle$ where X is the set of input values and Y is the set of output values, S is the set of sequential states, δ_{int}: $S\rightarrow S$ is the internal transition function δ_{ext}: $Q\times X\rightarrow S$ is the external transition function, λ: $S\rightarrow Y$ is the output function, $ta: S\rightarrow\mathbb{R}_0^+$ is the time advance function, $Q=\{(s,e)|s\in S, 0\le e\le ta(s)\}$ is the set of total states, e is the time elapsed since last transition.

Every atomic model can be coupled with one or several other atomic models to build a coupled model. This operation can be repeated to form a hierarchy of coupled models. The set of atomic and coupled models and their connections are named the structure of the model. This leads to the following notation:

$DEVS_N=\langle X,Y,D,EIC,EOC,IC\rangle$

where X and Y are input and output ports, D the set of models, EIC, EOC and IC, respectively, input, output and internal connections. Moreover, DEVS is an operational formalism, i.e. it provides the algorithms (the abstract simulators) that implement the formal models. So, since the beginning of the DEVS works, several DEVS simulators are implemented. The next section develops the VLE simulator, based on the DEVS formalism.

3.1.2. VLE

VLE (Quesnel et al., 2009, 2007, *Virtual Laboratory Environment*[1]) is a software and an API (*Application Programming Interface*) which supports multimodeling and simulation by implementing the DEVS abstract simulator. VLE is oriented toward the integration of heterogeneous formalisms. Furthermore, VLE is able to integrate specific models developed in most popular programming languages into one single multimodel. VLE implements the dynamic structure discrete event (DSDE) formalism (Barros, 1997) which provides the abstract simulators for parallel DEVS (PDEVS) (Zeigler et al., 2000) for the parallelization of atomic models and dynamic structure DEVS (DSDEVS) (Barros, 1996) for the M&S of systems where drastic changes of structures and behaviors can occur over time. DSDE abstract simulators gives to VLE the ability to simulate distributed models and to load and/or delete atomic and coupled models at runtime. VLE proposes several simulators for particular formalisms; for instance, cellular automata, ordinary differential equations (ODE), spatialized ODE, difference equations, various finite state automata (Moore, Mealy, UML statecharts, Petri-nets, etc.) and so on.

This framework can be used to model, simulate, analyze and visualize dynamics of complex systems. His main features are: multimodeling abilities (coupling heterogeneous models), a general formal basis for modeling dynamic systems and an associated operational semantic, a modular and hierarchical representation of the structure of coupled models with associated coupling and coordination algorithms, coupling of pre-existing models, distributed simulations, a component based development for the acceptance of new visualization tools, storage formats and experimental frame design tools, and free and open source software.

3.1.3. Petri-nets

In order to include the Petri-nets formalism in VLE, the DEVS approach is applied to the Petri-nets (Peterson, 1977). Works dealing with the mapping of Petri-nets into DEVS exists (see Jacques and Wainer, 2002 for instance). In these works, places and transitions are specified as atomic models and the network as coupled model. In our approach, a Petri-net simulator is wrapped in a DEVS simulator. In the following paragraphs, we give the definition of a Petri-net: $PN=(P,T,F,W,m_0)$, where:

$P=\{p_1,\ldots,p_p\}$ is a set of places and $p=card(P)$;
$T=\{t_1,\ldots,t_t\}$ is a set of transitions and $t=card(T)$;
$F\subseteq(P\times T)\cup(T\times P)$;

[1] http://www.vle-project.org