Engineering Applications of Artificial Intelligence 23 (2010) 830–843

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Hybridation of Bayesian networks and evolutionary algorithms for multi-objective optimization in an integrated product design and project management context

Paul Pitiot [a,c], Thierry Coudert [a,*], Laurent Geneste [a], Claude Baron [b]

[a] Laboratoire Génie de Production, Ecole Nationale d'Ingénieurs de Tarbes, 47 av. d'Azereix, BP 1629, 65016 Tarbes, France
[b] Laboratoire Toulousain de Technologie et d'Ingénierie des Systèmes, INSA de Toulouse, 135 av. de Rangueil, 31077 Toulouse, France
[c] Centre de Génie Industriel, Ecole des Mines d'Albi, Université de Toulouse Campus Jarlard, 81013 Albi CT Cedex 09, France

ARTICLE INFO

Article history:
Received 18 June 2009
Received in revised form
13 November 2009
Accepted 16 January 2010
Available online 10 February 2010

Keywords:
Project management
Product preliminary design
Evolutionary algorithm
Experience feedback
Bayesian network
Learning

ABSTRACT

A better integration of preliminary product design and project management processes at early steps of system design is nowadays a key industrial issue. Therefore, the aim is to make firms evolve from classical sequential approach (first product design the project design and management) to new integrated approaches. In this paper, a model for integrated product/project optimization is first proposed which allows taking into account simultaneously decisions coming from the product and project managers. However, the resulting model has an important underlying complexity, and a multi-objective optimization technique is required to provide managers with appropriate scenarios in a reasonable amount of time. The proposed approach is based on an original evolutionary algorithm called evolutionary algorithm oriented by knowledge (EAOK). This algorithm is based on the interaction between an adapted evolutionary algorithm and a model of knowledge (MoK) used for giving relevant orientations during the search process. The evolutionary operators of the EA are modified in order to take into account these orientations. The MoK is based on the Bayesian Network formalism and is built both from expert knowledge and from individuals generated by the EA. A learning process permits to update probabilities of the BN from a set of selected individuals. At each cycle of the EA, probabilities contained into the MoK are used to give some bias to the new evolutionary operators. This method ensures both a faster and effective optimization, but it also provides the decision maker with a graphic and interactive model of knowledge linked to the studied project. An experimental platform has been developed to experiment the algorithm and a large campaign of tests permits to compare different strategies as well as the benefits of this novel approach in comparison with a classical EA.

1. Introduction

Many companies, in order to meet the requirements of their clients and to provide them with adequate products, implement two key processes:

- the "product design" process, which aims at defining precisely the components and the structure of the product,
- the "project design" process which aims at specifying how the product will be realized (sequence of tasks, used resources…).

These two processes are often implemented sequentially: first the product is designed then the realization project is elaborated. For example, when a client wants to build a house, the architect designs at first a plan of the house, then the corresponding realization project is developed and launched. Since the project constraints (for example resources availability) are not explicitly taken into account in the product design, this can lead to additional iterations between "product design" and "project design" processes. A better integration (or coupling) of both processes is therefore a way to improve the global performance of companies.

An in-depth study of several mechanisms that can facilitate this integration has been launched in a project called ATLAS, funded by the French National Research Agency and involving academic laboratories, industrialists and the competitiveness cluster Aerospace Valley. The work presented in this paper takes place in the context of the ATLAS project.

In this section, a simplified product/project integration model is proposed. Indeed, in both environments (product and project), design processes are generally achieved according to a

* Corresponding author.
E-mail addresses: paul.pitiot@enit.fr (P. Pitiot), thierry.coudert@enit.fr (T. Coudert), laurent.geneste@enit.fr (L. Geneste), claude.baron@insa-toulouse.fr (C. Baron).

0952-1976/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2010.01.019

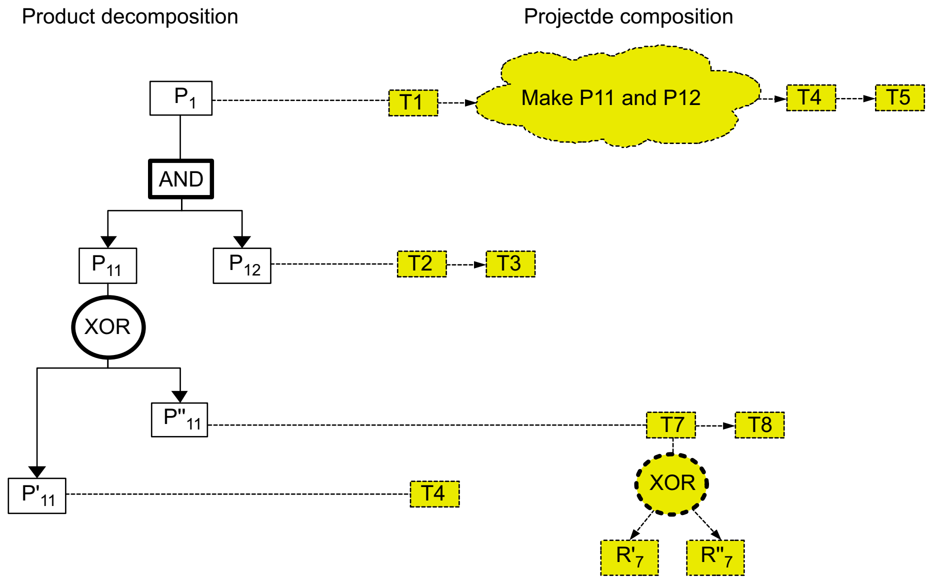

Fig. 1. Example of product/project decomposition.

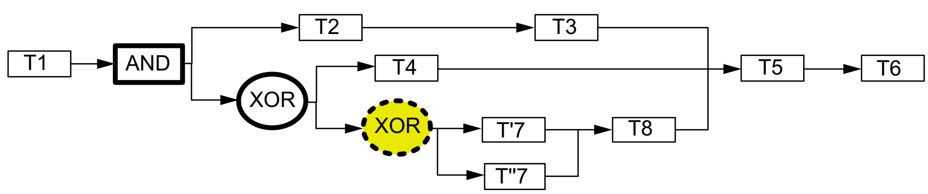

Fig. 2. Example of integrated model: the project graph.

hierarchical decomposition (see Fig. 1) in order to encompass complexity:

- products are recursively decomposed into smaller sub-products ("AND" connectors), e.g. product P_1 is made of P_{11} and P_{12},
- accordingly, projects are recursively decomposed into sub-projects: in order to realize P_1 one has to realize P_{11}, to realize P_{12} and to assemble P_{11} and P_{12},
- alternatives ("XOR" connectors) can be defined in products (e.g. choice between components P_{11} is composed of P'_{11} XOR P''_{12}) and in projects (e.g. choice between sub-contractors R'_7 XOR R''_7 to achieve task T_7).

Definition 1. an integrated model, called **project graph**, is used in order to represent simultaneously the links between the product and project hierarchies. This model consists in an oriented graph without cycles in which nodes are: tasks of the project, "AND" connectors and "XOR" connectors. The oriented arcs represent the precedence constraints between tasks. Fig. 2 represents such a model for the example of decomposition given in Fig. 1. Such a graph permits to capitalize that is called "structural knowledge" in the rest of the article. It concerns XOR nodes that correspond to the possible choices of products' structure. Making a product choice corresponding to a XOR node imply to inhibit a set of downstream connected nodes. Those product XOR are represented by a circular node whereas project XOR, which do not involve inhibition of other XOR node, are represented by dotted circle.

Definition 2. a **scenario** corresponds to a graph in which all the choices are made (i.e. with no more XOR nodes). An example of scenario, corresponding to the model in Fig. 2, is illustrated in Fig. 3.

1.1. Mathematical description of the addressed problem:

The problem addressed in this paper consists in searching an optimal scenario among all the possible ones within the simple directed graph project. The project graph $G=(\alpha, \beta)$ is defined by:

$\alpha=\{\alpha_k\}$, the set of all nodes,
$\beta=\{\beta_{ij}\}$, the set of directed edges between the node α_i and the node α_j with $|\beta|$ the total number of edges.

The following subsets permit to formalize the problem defining the three different node types (XOR, AND, Task):

$T=\{\alpha_p/\alpha_p$ is a task node, $\alpha_p \in \alpha\}$ is the subset of task nodes with $|T|$ the total number of task nodes,
$XOR=\{\alpha_q/\alpha_q$ is a XOR node, $\alpha_q \in \alpha\}$ is the subset of XOR nodes with $|XOR|$ the total number of XOR nodes,
$AND=\{\alpha_r/\alpha_r$ is an AND node, $\alpha_r \in \alpha\}$ is the subset of AND nodes with $|AND|$ the total number of AND nodes.

Let X, the vector of discrete variables (decision variables) corresponding to XOR nodes

$$X=\{x_i/\alpha_i \in XOR\} \quad (1)$$

Let Dx_i, the domain of the variable x_i defined by the vector of identifiers of the direct successor nodes of the XOR node i.

$$Dx_i=\{j/\alpha_j \in \alpha, \beta_{ij} \in \beta, \quad \alpha_i \in XOR\} \quad (2)$$

A decision associated to a XOR node α_k that participates to a scenario s corresponds to the instantiation of the variable x_k and is defined by

$$x_k^s=j \quad \text{with} \quad j \in Dx_k \quad (3)$$